DRAGON USER

The independent Dragon magazine

October 1988

Contents

Editorial

I WAS just sitting here thinking what a nice lot you Dragon lot are. I can't think of more than three people I never want to hear from again in all the time I've been sitting at this desk(s). What is it about the Dragon that brings out the best in its ops? A lovable personality? Protectiveness? A conviction that there is always one more thing you can do with it? Cheapness? Who knows? I said that Dragon users move in a mysterious way, like God. Take it from me, it beats dealing with IBM.

Thanks to the folk who wrote to say that they had substantial collections of DU. I'll expand on what was on my mind in next month's Letters page. I once met someone whose colleague's dad had worked at Dragon Data. He couldn't remember the address. Now I wish he could.

The time has come to say again: don't forget the Show. The Colour Computer Convention in Weston-super-Mare (Weston-in-Sea as it's known to its natives) on SUNDAY 4th December. Support your show, and it will support you. I'm glad to hear that nearly all the display space is booked out. This must be a good one.

Pete Gerrard is on holiday

Telephone number
(01) 570 8335

Editor
HELEN ARMSTRONG

Production Editor
HELEN ARMSTRONG/ARTSET

Administration/Advertising
BOB HARRIS

Publisher
DRAGON PUBLICATIONS

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues

ISSN 0265-177

Address: Dragon Publications, 49 Alexandra Road, Hounslow, Middlesex TW3 4HP, United Kingdom.

Published by Dragon Publications 1988

Typeset by Artset Limited, London NW1

Printed by Headley Brothers Ltd. Ashford, Kent

Registered at the Post Office as a newspaper. Dragon and its logo are trademarks of Eurohard Ltd.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters 49 Alexandra Road, Hounslow, Middlesex, TW3 4HP

Cure for hiccups

JONATHAN Cartwright tells me that Brian O'Connor has written to you with information about the 'space hiccup' (*Yes. See below*) in *Utopia*. Reference was made to it in the instructions, but what it was and what it did was not entirely clear. Jonathan later explained it to me exactly, but too late for the review.

Because of this a small part of the review is not entirely true: observe the 6th paragraph on page 19: 'Should your fuel fall below zero ... you will die and be sent back to the beginning of the game.' That's not quite true. The game is split into five stages. Once you enter a new stage, you should hear a small beep. From then on, if you are killed, you start at the beginning of the new stage (if you have any lives left). So my statement that it is unfair is not justified.

I do apologise for any inconvenience. The rating and all other statements still stand.

If you're looking for someone to do the Expert for a month or two, I'm available.

Donald Morrison
72 Diriebught Road
Inverness
IV2 3QT

I hear a small beep ... if I die now, I get to eat my supper again. We are up to date with surrogate Experts, thanks, but if a vacancy arises be sure I will contact you first, the noo. Bob is I hope seeing to your other queries. No, you aren't sacked. Anyone can make a mistake, as the Dalek said to the dustbin.

Every month we will be shelling out a game or two, courtesy of our supplies, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

Today the Dragon...

I owe a lot to the Dragon because I knew nothing about computing before I bought it, but I have learned how to program in Basic and Forth, and have gained wide experience of word processors, spreadsheets, databases etc. which has held me in good stead when computers eventually arrived in my office, an advantage which I still hold over my fellow workers.

I enclose reviews of two packages from Compusense, *Edit +* and *Dynafast*. Neither of these is new but they are still available and have not even been mentioned by your magazine for a considerable time. They may be of interest to newer readers.

Clive G. Scott, 62 Berry Meade
Ashstead, Surrey KT21 1SG

AHA! So you are the mystery reviewer. I've been looking for your covering letter everywhere, and here it is in my file labelled 'Letters'. Who would have thought it?

I will try and find out whether the two programs are still being supplied by Compusense and what the prices are, but I agree that they should be drawn to newer users' attention.

I think it is unecessarily pessimistic to dwell on what-might-have-been for the Dragon, so I won't print that part of your letter. We've all been through it in the dark of the night (and on the letters page, in bits and pieces). Regard what Paul never tires of saying: many users 'upgrade' to something with 'more Ram' and, unless they have bought the machine for some specific talent, they are often none the wiser. At least you have used your Dragon to learn about computing.

It's just a passing thought, and I suppose money has a lot to do with it, but why discard your old computer when you buy a new one? OK, if you have spent your life savings, and you want to change over a year later ... or if you are really short of funds ... but the sale of a Dragon these days won't help you far towards any other machine worth purchasing, and if you have bothered to exploit your Dragon to the full, even in an unexpanded version, you are discarding a lot of knowledge and experience if you don't maintain it on the sidelines.

Listen, I know a guy who recently paid £150 — I would put that in gold stars for emphasis if I could — £150 for a complete second hand Greengate sound 4-track sampling system, complete with Apple 2e, dual disc drives and synth keyboard. OK, the Apple 2e in that form is a dinosaur, but think of the power you have there ... I would have killed him to get at it, and so would most of his colleages. But there are bargains like that around if you keep your eyes open.

Your ex-Dragon, gentle reader, could be someone else's dream bargain. Think about that twice and hang about ...

One last thing ... I know it's traditional to address editors as Sir, but I've been with DU over two years now, and the only person who calls me Sir is my husband, and he doesn't get away with it.

We're getting a good calibre of letter this month.

Back to Square Five

WE would like to thank Donald Morrison for his excellent review of our arcade game *Utopia* in DU August 1988, but would like to point out a slight inaccuracy.

Mr. Morrison says that the player restarts at the beginning of the game when a life is lost, but in fact the game 'map' is split into five sectors, and if you reach a new sector and lose a life, you return to the start of that sector, *not* the start of the game.

He may have been reviewing a prototype of the game, as this feature was not added until late in the game's development. We agree that when this shortcoming *does* occur in games it is very annoying, and we asked Jonathan Cartwright to include this feature in *Utopia* before it was released, just for this reason.

Neil O'Connor
Pulser Software
36 Foxhill
High Crompton
Shaw
Oldham
OL2 7NQ

HA! They've all gone soft .. in my young day you got sent back to the beginning of *Asteroid Storm* even if you were on your 93rd screen. And it wasn't even meant to have 93 screens.

Dynamic Dream

REFERENCE Pam D'Arcy's comments on relocating *Dream*, I have an elderly Dream on cassette. This cassette loads Dream from hex 6C80. It then uses page hex 6B (eg hex 6B00 to 6BFF) for control fields. Note that this is the first complete page below hex 6C80.

Page 43 of the booklet that came with the cassette says that Dream is written in position independent code throughout, and so can reside anywhere in the memory map. It also says that it dynamically looks for the first complete page of Ram below itself and sets the Direct Page register (DPR) to point to it. This is consistant with the hex 6C80 loading. In fact, the booklet is not quite correct. If Dream is loaded from the start of a page (eg hex C000 or hex 6C00) then Dream skips a complete page. Thus for the examples given it uses page hex 7E (not 7F) and hex 6A (not 6B). Note that page hex 7F is the first

page of Ram under page C0 (which is Rom).

You can see that according to where Dream is loaded, you can get anything from one byte to 256 bytes of free space before the Dream workspace. This is important if you use *Alldream*, since Bug needs 128 bytes above the Dream space. If you load Alldream to start between byte 1 and byte hex 7F or a page (eg hex 6001 to hex 607F) then if you use Bug break points you might overwrite Dream. Result chaos! But Dream without Bug is safe.

Pam is correct in her guess about why Dream does not work for Dragon 64K Ram mode. It would be trivial to alter Dream to work for just the 64K mode, but I don't suppose it's worth the trouble.

About pages: Dragon machine code when reading or writing data can either 'see its paper' as one vast sheet, or as a book of pages. In this 'direct' mode the memory is divided into 256 'pages' each containing 256 bytes. Both pages and bytes are numbered from 0 to 255 (hex FF). The page that the Dragon sees depends on the value (0 to 255) that is loaded into the DPR.

When in direct (page) mode Dragon needs less bytes of code and runs faster, but it can only access 256 bytes. However, it only takes 12 microseconds to change the page.

Dream uses Direct Page when handling its control fields.

Ted Newman
113 New Haw Road
Addlestone
Surrey KT15 2DA

Hack and Poke?

ON the subject of (commercial) programs, maybe someone could help me with David Maken's *Picture Maker*? I can get the screen dump to work, but it loses, or rearranges, parts of a screen, especially when told to dump a colour screen as opposed to a black and white (Pmode4) screen. Also can anybody hack it and give me a poke to put it into Pmode4? I've tried, but only hacking program B will touch it, and then it only loads a small segment which consists of an On Error Run and some other incomplete code.

On the splitting of the Dragon world (Tandy/Dragon, 32/64 etc.) what about those of us who only use discs or tapes? Whereas it may be easy (alright, it's not, but to keep it in context) to convert a program to disc, how can someone who doesn't know anything about discs even attempt to convert one for tape operation? I am of course referring to programs printed in these erstwhile pages.

Some advice for those of you with Dragon Data joysticks who have 'broken' them in some way or other. The top half of the casing must be prised away from the bottom half to gain access (difficult). Also Maplin switch JB00A on P444 of the current catalogue is a good fit for replacing the fire button ... even if it does tend to melt at 25 watt soldering power!

For those of us who wordprocess on Peter Whittaker's *Wordprocessor*, pressing <SHIFT> + one of the left/right arrow keys enables you to jump through your text from CR to CR while in the editing mode. And it is also necessary to clear the glossary when starting from scratch before entering any words. I cannot speak for the disc version, but suspect that these hints are correct for disc users as well. Anyhow, judging from the number of letters on Peter's word processor since it was published, perhaps we ought to start a P. Whittaker's Wordprocessor User Group!

To finish up, how about shelling out rebate coupons instead of games for the best letter of the month? So that serious users can get themselves some more (serious) utilities?

Gareth Sims
24 Logwell Court
Standens Barn
Northants
NN3 3TN

NOT a totally silly idea, that. I'll look into it. Incidentally, one of the great unsolved mysteries in the History of the Dragon is: what happened to Peter Whittaker? We lost track of him out Ely way somewhere.

M1009 to HR5?

YOUR *Down in the Dumps* article in November 1987 came near to my requirements, but not near enough. I have a Brother HR5 printer. Can anyone tell me how to modify the M1009 program for use with the HR5? I presume all I need to alter are the EXC codes in EFDC to EFEC. Your help would be appreciated.

Keep up the good work.

R F Roach
Trenellan
Manaccan
Helston
Cornwall
TR12 6HY

In heaven with Lucifer

I have owned a Dragon 32 since September 1983. In this time good and bad (mainly good) games have been produced. Every so often another game is hailed as the best game ever. Well, its now time to move 'the best game ever' tag held by *Shocktrooper* to its new and rightful place on *Lucifer's Kingdom*.

After recently purchasing this incredible game from Orange Software (they deserve a mention) (*Listen, mate, they're so efficient you cant miss them.*) I loaded the cassette into my Dragon 32 (the family computer)(that's what it says on the box. How come the family portrayed aren't entangled in a thousand wires? You should see the state of my bedroom.) On loading *Lucifer's Kingdom* (which is a stupid name for it) you are presented with a screen which I can only describe as XXXX (you may have to put 'very poor') — mouldy orange and lilac characters. Still, you can't judge a book by its cover.

The game starts automatically and you find yourself controlling a hovering, flying thing which fires two missiles when you press the left joystick button. The game is in PMODE 4 and is quite honestly the best game I've played. It's fast, smooth, addictive and very frustrating. After hours of play and aching fire button thumbs I got to the second of five regions. Only another 20-odd planets to conquer. It was because of July's *Dragon User* review that I bought the game.

I find it reassuring at this time of the Dragon's life to encounter brilliant software and it's because of this and the continuing quality of *Dragon User* that I've decided not to trade my Dragon for an Amstrad (spit) and to re-subscribe for another year.

All this praise from a man whose full-time occupation is working with ICL and IBM mainframes. Have you ever tried writing a shoot-em-up program in Cobol?

D J Platt
46B Winchester Road
Burnham-on-Sea
Somerset
TA8 1HY

HAVEN'T you heard of our new publication Cray Gamer? Our first issue will feature a cute shoot-em-up where the player simply takes out Saturn and Jupiter. Pity only one person can play it!

Hi score corner

I would be very much obliged if you were to print the following message in your magazine. It is about a new *Chuckie Egg* high-score ...

Now, before I give you my hi-score, I would like to say a few things to Andreas Verwegen, who scored an 'unbelievable' hi-score of 15,094,650! Well. Andreas, you have been beaten into a SMALL second!!! My hi-score on *Chuckie Egg* was around 52,000! No, wait a sec ... was it 160,564,969? Or was it 294,785,289? Or was it 368,102,689? NO! It must have been 714,732,200!!! I lost count of the level, but according to Andreas' score/level ratio, it must have been about 38,780!!!

Mark Henegan (age 15)
2, Orchard Close
Biggleswade
Beds SG18 0NE

PS If anyone beats my score, WATCH OUT!

SMELLS fishy to me, this ... he says he's 15, but his spelling's better than mine and, according to my calculations, he has spent approximately 73 years obsessively playing *Chuckie Egg* ... is he (a) Dr. Patrick Moore (b) Professor Sir Randolph Quirk (c) Dr. Isaac Asimov? Answers on a postcard, please, along with your current top five software titles, to the usual address. You guessed it, I'm thinking of reviving the People's Chart - probably on an irregular basis (gives the new product more time to hit the streets). Why should *Chuckie Egg* have all the fun?



If you have any new products for the Dragon — software or hardware — please write to The Newsdesk at 49 Alexandra Road, Hounslow, Middlesex, TW3 4HP

New and converted from Dragonfire

DRAGONFIRE Services have a limited number of *The Tape Doctor* cassettes, which were published by Computil, for the Tandy CoCo 32K ONLY. The program, which helps users recover bad files from tape, received a five-Dragon review in *Dragon User* when it was released. Priced £4.00 plus P&P, the tapes have colour covers and are available while stocks last only.

New from Dragonfire is an adventure called *Hole* by R E Hemmings. Your ship is sucked into the infamous Black Hole. What happens next is not told ... price £3.00 plus P&P.

New on disc is *Printer Prompt II*, Dragonfire's 'electronic typewriter' program, compatible with DragonDOS and CumanaDOS 2.0. The price is £4.00 as for the tape version. The program has a 64-character screen display, is menu-driven and is compatible with Epson and Epson-type printers.

Dragonfire is currently converting many of its programs to disc, initially for DragonDOS and CumanaDOS. Dragonfire would like to hear from users interested in conversions to DeltaDOS so that they can assess demand.

The company is working on a major addition to its list for the Colour Computer Convention in December.

For more information please send a stamped SAE to Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent NP3 3NH. Postage is 50p per tape UK, £1.25 overseas.

H C Anderson list

H C Andersen has published a new product/price list for the Dragon which is available from them at Englandvej 380, DK-2770 Kastrup, Denmark. The list is a 16-page A5 booklet and includes hardware and discs, upgrade kits, spare parts, OS-9 and Flex software, games and utilities.

HCA are the European licensees for OS-9. Prices in the English-language list are quoted as for export in sterling (GBP) ex VAT. Enquiries about VAT and customs duties should be addressed to HCA.

New Era expands

Simon Jones's New Era Publications are issuing two booklet guides to Dragon products and suppliers. The *Dragon User's Handbook* 'details every consumable currently available for the Dragon (and) every company/user group currently supporting the Dragon' according to Jones, who adds that they believe this will help in the fight against piracy by identifying software available legitimately. The price is £1.65. The *Dragon Directory* is a bi-monthly update which 'is aimed at keeping users in contact with other users/companies (and) will detail any new software which has been released prior to publication of the directory'. The price is £4.95 for a two year subscription. New Era's present subscribers are entitled to a discount of approximately 15% on both booklets.

New Era has recently taken over the list (undetailed) of Unique Software, who were publishing Dragon software in 1984, and hope to revive other deleted software in the future.

Enquiries to New Era Publications, c/o Simon Jones, 37 Collins Meadow, Harlow, Essex CM19 4EN

Lee goes American

Gordon Lee's competition page featured (through 'Winners and Losers') in the July 1988 edition of *Scientific American* when Gordon issued to readers of that August journal the challenge he issued to us in the July 1988 *W&L* - to construct a square of 6 by 6 digits which contained more than 170 prime numbers. The Computer Recreations column offered some tips for constructing such a square, and offered to print any solutions which beat Gordon's.

We await the outcome. If anybody out there cracks it — tell SA as well as us, and tell them where you come from! We await the outcome.

Update to date

THE July 1988 edition of *Dragon Update*, the Newsletter of the National Dragon Users Group, contains a report from the Ossett Show, reviews of *Lucifer's Kingdom* and *Utopia*, a look at the CoCo 3, news, programming articles, and two appeals: one for local area members to run the NDUG stall at the Weston super Mare show in December, and the other for somebody to help Paul Grade get a backup photocopier from Lincolnshire to Worthing.

The National Dragon User Group is a user group offering technical assistance, a forum for Dragon and Tandy users and 12 newsletters a year. For information write to Paul Grade, 6 Navarino Road, Worthing, Sussex.

Orange across the Sea

Orange Software is releasing an adventure by Geir Hovland, *North Sea Action*.

This is a pacman-type game in which the player sends a drill down below the North Sea in search of oil and oil drums. 'Not the most intellectually testing game, but it has that "one more try" quality' says Graham Smith of Orange.

Orange's brochure is now a nice little A5 booklet. Write to Orange Software, The Garth, Start Road, Nant-y-Derry, Avergavenny, Gwent NP7 9DP for information.

Starship rolls again

THE next release from Starship Software will be *Impossiball*, featuring full colour perspective graphics, music and digitised speech. The game is being finished and more details will be released shortly.

Impossiball will be published by Pulser Software on cassette or DragonDOS disc for the same price. Pulser are also planning 'a new text adventure, a new self-definable database program, and a computer aided design program, *CAD 6809* (probably priced at £4.99) in September.

Discounts of around 10% on many of their older programs will be available. Enquiries to Pulser Software, 36 Foxhill, High Crompton, Shaw, Oldham OL2 7NQ.

Dragonsword!

Paul Grade gets his inspiration from a fruit.

HEARD about the new Official Secrets Act yet? Don't worry, it isn't likely to apply to anything you're likely to write on your Dragon, at least not unless you're really working for MI5 and trying to discover the inside secrets of *Dragon User.*

A lot of programmers appear to imagine that whatever they write is an 'Official Secret', especially in the Dragon world (*Not just the Dragon world, snarls your Editor-about-town*) and spend more time writing protection routines than they do writing programs. What's all this about? Well, I'm getting a little tired of hearing all about the Dreaded Pirates who spend all their time copying programs and passing them around, thus depriving poor hard-working programmers of thousands of sales. It may well have been true once, and there are still a few morons around who imagine that buying one copy of a program between 99 people is a good idea, but if someone is really determined to break into a protected program they will, so what on Earth is the point of writing all these elaborate routines? Personally I was always told that the first thing one should do when buying a disc or tape was make a backup copy and keep the original in case of accident. That makes a lot of sense to me, and if I have to spend several hours fiddling about before I can make a backup copy of a program then I'm simply not going to buy the thing in the first place. I don't care how good a program may be, or what a bargain it is, I insist on the right to be able to make a backup copy, and if this involves having to break a protection routine then I refuse to buy it.

The whole thing is ridiculous. I KNOW sales are low now, but that has nothing to do with 'pirates', it's simply that there are a lot fewer Dragon owners now than there were, and those that remain tend to be more interested in using their machines than running the latest games software. When the programming world realises this we might get somewhere, and more to the point they might even get a few more sales.

As well as the Dragon I have to admit to running an Apple 2+, which as you will probably know is an equally antiquated machine, but has the advantage of an established business and educational user base and thus a lot more good quality software than is available to Dragon owners. The programs aren't cheap, about eight times the price of Dragon software, for all you price-complainers, but very few of them are protected in any way, and to counteract those few that are there is a large selection of commercial copying programs. Almost every Apple disc, regardless of source, contains an instruction to make a backup and NOT use the original, and several firms throw in a batch of freebie programs and routines as well as the main program, plus summaries of forthcoming releases. When was the last time you found anything like that from a Dragon software firm? It is a much better attitude, and has to be more attractive to potential customers than the paranoia about 'piracy'.

I am not defending the real pirates, those people who copy a program, change a couple of lines and then try to market it as 'original work'. That isn't just piracy, it is also theft, and there is no excuse at all for it. Likewise, I hold no brief for 'software libraries' who make copies of commercial software and rent them out in order to make a profit. The harder these types get jumped on, the better, but actual sales losses through private copying are minimal, and I think are probably no greater than sales lost through over protectiveness. Anyone selling fifty copies of a program in the first six months is doing 'as well as can be expected' now. The market just isn't there any more, and more to the point, Dragon users generally now know a lot more about their machines than in the 'Good Old Days', and are quite capable of writing simple programs for themselves, which means that a commercial program has to be good in order to sell at all.

I'm not having a go at programmers. Generally they do their best to provide good material at reasonable prices, and certainly none of them are going to get rich out of it. But a few of them just refuse to believe that there is now a limited market, and that games sales are so low that they'll be lucky to cover the cost of the listing paper they used in writing the program. Blaming piracy is self delusion. Those of us left need the software, and are willing to buy it. There is still a market, but it isn't big enough for anyone to make a profit from, so what's the point in expecting one?

If anyone thinks I spend all my time complaining, they're wrong, I never spend more than 75% of it that way, but if you want to know why, try starting a user group or a magazine. Well over half the letters that come in contain at least one complaint about something or other, usually things over which I have no influence at all, and the list of "Why doesn't someone market/write/publish/explain/invent etc.' would fill a dozen issues of this magazine easily. (Now that's an idea Boss Lady hasn't thought of yet. I think. (*A sort of whinge corner, you mean? I thought the Letters Page went far enough, but now you mention it ...*) All I am really doing is replaying a few of the comments and 'whys' to you, hopefully for action, but failing that a few solutions would help. Of course I add a few questions and comments of my own, but I'd be a fool to miss the chance, wouldn't I?

On the subject of questions, there's one I really must ask. It may not be of quite such importance as the meaning of Life, the Universe, and Everything (and anyway, we all know the answer to that one), but it is a question which has caused generations of Editors and writers to leave claw marks in the wallpaper and increase the value of Distillers Company shares by several thousand percent. Is there anyone out there? Personally I don't believe that any of you are there at all, you're just something out of a very prolonged dream, but if there really is life on the other side of the keyboard just knock three times for yes, twice for no. I mean, if there really was intelligent life out there why would I get letters telling me that the writer has been a subscriber to *Dragon User* for several years but has only just heard about the *National Dragon Users Group*, from a friend? Why would I get letters from Group members asking me if *Dragon User* still exists? I've been giving *User* mentions in *Update* for years now, and at one stage even carrying *User* advertising and subscription forms. *Dragon User* has been mentioning the Group in almost every issue for about the same length of time, so is there really someone out there reading this, or are you all a computer malfunction? (But if you are, who is sending in the subscription cheques and the letters full of questions I can't answer?)

Enough of this rubbish. I'm being paid (eventually, I hope) to write serious, profoundly interesting material for this prestigious magazine.I know they are intelligent because I read Pete Gerrard's page too, so I can learn all about how to do deals with trolls (very handy if I happen to get stuck in a Scandinavian myth), fairies (useful in a couple of pubs I know), wizards (are they the ones responsible for sunny spells?), and similar everyday phenomena. The bit I don't understand though is why Pete's regular reader(s) are so hooked on the now traditional Adventure scenario. Tolkien has a lot to answer for, I know, but surely someone could break out of the pattern? How about an Adventure program based on parking a car in London or Brighton? The hazards make mere trolls and demons seem positively benign. Or how about one for all ex-Army and Rugby Club members, based on the Story of Eskimo Nell? So you must have wizards? Try writing a Spell Casting Utility (with a special sub-menu for putting curses on people who offend). Of course, there is a slight snag with that one. If it worked properly it could result in several people I can think of having to hop rapidly to the nearest pond, and our Illustrious Editor might have a harder time than usual parking her broomstick (*Please. I ride a Vax.*) but who cares? Let's have a bit more variety. Please?

Paul

New software for review should be sent to *Dragon User*, 49 Alexandra Road, Hounslow, Middlesex, TW3 4HP

Come back, Compusense!

Program: *Edit +*
Supplier: Compusense, PO Box 169, Green Lanes, Palmers Green, London N13 5TN
Price: Unknown. Cartridge or disc.

THIS product comprises the screen utility *Hi-res*, which is also available on its own and *Edit +*, a full screen editor for Basic programs.

Hi-res uses a special character set to produce a 51 x 24 text display on a mode 4 screen. The display, which is similar to that of the word processor *Telewriter*, can be displayed in black text on green or buff backgrounds or the inverse.

A PRINT ! in place of the normal PRINT @ command, which can still be used for a 32 x 16 display, gives access to all 1224 print positions. Various foreign character sets can be selected by extensions to the CLS command. Text can be freely mixed with graphics. *Hi-res* is not as versatile as *Rainbow Writer* and when combined with *Edit +* consumes considerably more memory, but it is worth it for the screen editor which is a vast improvement over the Dragon line editor. Type EDIT and the screen will clear and any Basic program in memory will be displayed.

The CLEAR key is used in combination with other keys to select the various options of the program. Auto repeat is implemented for easier movement around the screen, which is by means of the cursor keys. The screen unlike a word processor will not scroll when the cursor reaches the bottom of the screen but it can go up or down a screen at a time or it can jump to a specified line.

The default mode is overwrite but although insert can be selected it will return to overwrite when an alteration is confirmed by pressing ENTER. Inserting into a long line can be slow so it pays to extend the line first. If too much text is entered into a line it will beep and show an overflow marker. New lines can be added or old lines deleted.

Characters can be deleted individually at the cursor or from the cursor to the end of the line. Single or blocks of lines can be copied by placing markers and inserting into the required place in the program or moved by deleting the old block after the copy has been made. Markers will remain for further copies until removed.

The program can be searched for a specified string of characters which can be selectively replaced by an alternative string. This is ideal for changing variable names.

This covers the main features of the program which I find indispensable when developing programs. This is a must for the serious programmer.

Clive G. Scott



Program: *Dynafast*
Supplier: Compusense, PO Box 169, Green Lanes, Palmers Green, London N13 5TN
Price: Unknown. Cartridge or disc.

Dynafast comprises three programs: *Dynafast*, a Basic compiler, *Dynamiser*, a program compactor and *Dynaref*, a cross reference utility all selectable from an opening menu.

Dynafast compiles a basic program into a mixture of machine code and Basic, which, while not as efficient as a program written in assembler, runs much faster than a Basic program.

To gain the maximum speed advantage all integer variables should be declared at the start of the program by naming the variables and the location in memory for them to be placed ie @30000 A B C.

There are four alternative modes of compilation Normal, Fast, Step and Print. Normal mode will display the progress of the compilation but slowed down so that it can be read, Fast processes it as fast as possible, Print directs all output to the printer while Step mode will stop and wait as each variable/line is processed.

Compilation is done in 3 passes. Pass 1 displays the integer variables that have been declared, Pass 2 processes each line and indicated whether it was able to fully or partially compile the line and Pass 3 is a tidying up process. Compilation can be paused or aborted any time.

The compiled program can be automatically saved under a specified name but the START, END and EXEC addresses are displayed at the end of compilation and can be noted for future reference. The compiled program can now be run by typing EXEC.

Dynamiser compacts a program in three passes by first deleting all redundant characters and spaces, secondly deleting all unnecessary REMs and lastly combining as many lines as possible into one long line.

In all three programs a progress report is displayed as each line is processed.

It is advisable that the compacted program be saved under a different name because it can be difficult to read and edit.

Dynaref runs through your Basic program and produces a table of all variables and constants together with the all lines they appear on which can be output to the screen or printer. The table can be restricted by setting limits in the EXEC statement ie EXEC:AZ will only produce a table of Basic variables.

It is invaluable for checking the variables that have been declared, particularly when modifying a program prior to compilation by *Dynafast*. This is a trio of programs which will compliment any user's collection of programming aids.

Clive G. Scott



Oranges and cream in Lucifer's land

Program: *Lucifer's Kingdom*
Supplier: Orange Software, The Garth, Star Road, Nant-Y-Derry, Abergavenny, Gwent NP7 9DP
Price: £5.99 plus 50p postage

IT'S been a couple of months since I've got round to writing a review but at least now that the ink is flowing again I've got something to rave about.

Orange Software are the company providing the 'raveable' item. It's only a few months since they appeared on the market but now they have a comprehensive list of titles, old and new, largely made up of text adventures. *Lucifers Kingdom* is an arcade adventure so it's perhaps unfair to compare it with the majority of their titles but I would say it's undoubtedly the cream of their produce.

The Kingdom of Lucifer is set deep in space not in the bowels of the earth as may be expected but the obvious idea of exterminating the old devil is still there. The way to do this is to fight your way to him by shooting down hoards of enemy vessels which speed down the screen at you. The foe comes in various guises, some that you can dodge and forget, some that track you, some that move vertically down the screen and others that zig-zag. Whichever they are you never know what's going to come next and how many of them.

So far it sounds like a simple shoot-everything-in-sight type game. There is however a lot more to come than that. Firstly, you have to collect crystals not simply a matter of flying over them but shooting at flashing 'C' characters and gradually revealing them. This is not just to gather bonus points as in most games but a necessary part of the proceedings, because if you don't collect them you have to go back once the region is completed.

A region in this game is a set of set of six planets, a planet being a phase as, say, in

continued on page 8

Winners and Losers

Every month Gordon Lee will look at some prize programming

TWO for the price of one this month as we take a quick look at the April competition before turning our attention to the more difficult task of extracting the square root of 2.

When thinking about the April competition I said to myself "Why not make it an easy one this time to give everyone a chance?" And, sure enough, we were rewarded with one of the highest entries every with the poor old postman bent double beneath the weight of mail. The great surprise came upon opening them to find that well over half of the entries had arrived at the wrong answer. The reason turned out to be the wording of the puzzle which said (among other things) "There is a 9 among the cards but it is not the card one place clockwise from X". Now this was taken by many entrants to mean that there was only one 9 amongst the cards, whereas, in the solution that I had in mind, there were two nines present. This was not intended as a deliberate attempt to mislead, honest. By way of defense I can only quote the well-known puzzle:

Question: I have two coins totalling 55 pence. One of them is not a 50 pence. What are the coins?
Answer: 50 pence and 5 pence. (One of them, the 5 pence, isn't 50 pence.)

Anyway, using the information given, there were six possible sets of card values which agrees with the information given, of which the value of card X was 6 in three of the cases, 10 in two of them, and 4 in only one. The information that if you knew card X, you could determine the other values, clearly indicated that it is the last of these possibilities. The fact that the exclusion of more than one nine eliminated this possibility left many entrants with only five equally likely possibilities (some had even fewer, for various reasons), which resulted in the employment of much entertaining logic, and a fair amount of ingenuity.

A slightly more difficult task was the extraction of the square root of 2, to an accuracy of 125 decimal places. This was the competition for May, and, by contrast, the difficulty was reflected in the size of entries. But, congratulations to all those gallant readers who attempted this one. Incidentally, two errors crept in to the answer given on page 27 of the August issue. The 21st decimal place should be a 1 and not a 2 as given, and the 46th and 47th dps have been transposed and should read '37' and not '73'. (*Listen, I read that. I did. — Ed.*) Now it will work using the multiplication routine from last February's *Dragon User*. The method of computation outline in the 'Answer' section relating to this problem utilised a technique which was intended to prevent too much repetition while running the program. Unfortunately, this resulted in a rather complex listing — which would have been more of a hindrance than a help if it had been published. However, in response to requests for a listing which will do the job, here is one from D.J. Gray of Cleveland, shown here in his distinctive style of presentation. (When typing this in it is not necessary to include the spaces at the beginnings of the lines — these are there for clarity.) The Print #-2 is used in lines 1120 and 1130 to send the result to a printer. To display the result on screen these lines should be amended to just a plain PRINT command (but don't forget to include the semi-colon at the end of line 1130). The program will take about half an hour to compute the answer, but it will be given with absolute accuracy.

```
1ØØØ :    DIM A(126),Z(253),Y(253)
1Ø1Ø :    A(Ø)=1:Z(1)=1
1Ø2Ø :    FOR N=1 TO 126
1Ø3Ø :      A(N)=5:T=1
1Ø4Ø :      GOSUB 2ØØØ
1Ø5Ø :      IF T=1 AND Y(1)<2 THEN A(N)=9:T=Ø:GOTO 1Ø4Ø ELSE
T=Ø
1Ø6Ø :      IF Y(1)>1 THEN A(N)=A(N)-1:GOTO 1Ø4Ø
1Ø7Ø :      FOR I=Ø TO 2*N+1
1Ø8Ø :        Z(I)=Y(I)
1Ø9Ø :      NEXT I
11ØØ :    NEXT N
111Ø :    FOR I=Ø TO 126
112Ø :      IF I=1 THEN PRINT#-2,".";
113Ø :      PRINT#-2,MID$(STR$(A(I)),2);
114Ø :    NEXT I
115Ø :    PRINT#-2,""
116Ø :    END
117Ø '
118Ø '
2ØØØ :REM *******************
2Ø1Ø :REM *** CALCULATION ***
2Ø2Ø :REM *******************
2Ø3Ø :    FOR I=Ø TO 2*N+1
2Ø4Ø :      Y(I)=Z(I)
2Ø5Ø :    NEXT I
2Ø6Ø :    S=A(N)*A(N)+Y(2*N+1)
2Ø7Ø :    Y(2*N+1)=S-1Ø*INT(S/1Ø)
2Ø8Ø :    Y(2*N)=Y(2*N)+INT(S/1Ø)
2Ø9Ø :    S=Ø
21ØØ :    FOR J=N-1 TO Ø STEP-1
211Ø :      S=2*A(J)*A(N)+Y(N+J+1)+INT(S/1Ø)
212Ø :      Y(N+J+1)=S-1Ø*INT(S/1Ø)
213Ø :    NEXT J
214Ø :    Y(N)=Y(N)+INT(S/1Ø)
215Ø :    L=N
216Ø :    IF Y(L)>9 THEN Y(L-1)=Y(L-1)+INT(Y(L)/1Ø):Y(L)=Y(L)-
1Ø*INT(Y(L)/1Ø):L=L-1:GOTO 216Ø
217Ø :RETURN
```

Dragonsoft

New software for review should be sent to *Dragon User*, 49 Alexandra Road, Hounslow, Middlesex, TW3 4HP

Continued from page 7

Scramble. For instance Region 1 is Epsilion with six planets visited in the order Giotto, Maxima, Minima, Prolox and Lexita. That's Region 1. There are four other regions Delta, Gamma, Beta and Alpha making a total of 30 planets to navigate and with only five lives.

Particularly devious and infuriating obstacles for the gameplayer are static triangular land based objects which only allow your bullets past if pointing upwards, if they point down to get them up you have to fire at them. However after the next shot they reverse again. If you can follow that effectively 50% of your bullets get past but I can assure you that you struggle when your ammunitions fully flowing and to only have half can be disaster.

This game reminds me of *Vanguard* an old arcade favourite of mine, which shows how highly I rate this game. One factor that *Vanguard* had that this has, is addictiveness. At first it's a bit frustrating as you try and comprehend what's happening and how to play your role in it, but after a few plays you become more and more determined to reach that next planet, that next region with the next life of the next game.

There seem to be quite a few good new games about at the moment and this is certainly one of them. If you've got a joystick in need of exercise it's cheque in the post time, because to conclude with a totally abysmal pun this orange is a juicy one.

Philip Stott

DragonDOS Toolkit

D J Gray adapts the Premier Microsystems program for DragonDOS

FOR many people Premier Microsystems' *ToolKit* used in conjunction with the Delta Disc controller has been a very useful addition to the Dragon's facilities. *Toolkit* was designed to work with a Dragon 32 and a Delta DOS disc controller which was capable of containing an extra Eprom holding the Toolkit Editor. Those people who moved to a Dragon 64 would have found that their parallel printer did not work when *Toolkit* was activated and those who moved to Dragon or Super DOS found that they could not install their Toolkit Eprom.

The instruction manual supplied with *Toolkit* states that it cannot function correctly with Dragon Data's disc system as DragonDOS rather inconveniently takes the current video screen as a work area, thus defeating *Toolkit*, also that the Dragon Data disc cartridge contains no extra Eprom space for *Toolkit* to reside.

Those statements provide quite a challenge but it has been found possible to use a Dragon 64's extra Ram to hold *Toolkit* and with a DragonDOS cartridge attached make *Toolkit* operate. The problems to overcome were:

1) To obtain a copy of *Toolkit* that can be read into a Dragon 64's memory
2) How to summon the extra 32k of Ram of a 64 in order to store *Toolkit* in the correct position
3) *Toolkit* contains a self destruct routine that is activated if it is held in Ram; this has to be defeated.
4) *Toolkit* contains command words that are identical to some within DragonDOS; these have to be changed to prevent confusion.
5) How can *Toolkit* be modified to prevent it overwriting areas occupied by DragonDOS when using the CLS, FRAME and MOVE commands.
6) How can the system be modified to allow a parallel printer to be used when *Toolkit* is used with a 64.

Toolkit when installed in conjunction with DeltaDOS resides in memory between &HE000 and &HF9DF. This can be copied onto tape by using CSAVEM "TOOLKIT",&HE000,&HF9DF,&HE002. This copy can be used later to place *Toolkit* into the Ram of a 64.

Extra Ram

To gain control over the extra Ram in a 64 with a DragonDOS cartridge attached is quite straightforward (remember that it is not 64 mode that is wanted, only access to the extra Ram). **Listing one** is a routine that simply reads the information stored in Rom (Basic) and the DragonDOS cartridge then places it into Ram. This routine also modifies the RESET to ensure that if RESET is pressed then the system will not return to 32 mode. Later this routine is

```
LISTING 1

10 :REM ************************************
20 :REM *** LOADER TO PUT ROM AND DOS INTO ***
30 :REM ***         RAM OF A DRAGON 64     ***
50 :REM ************************************
60 :   FOR I=&H4E20 TO &H4E5A
70 :      READ A$
80 :      POKE I,VAL("&H"+A$)
90 :   NEXT I
100 :  EXEC&H4E20
110 :'
120 :'
130 DATA 8E,80,00,1A,50,B7,FF,DE,A6,84
140 DATA B7,FF,DF,A7,80,8C,DF,FF,25,F1
150 DATA 30,8C,1D
160 DATA 10,8E,03,EB,A6,80,A7,A0
170 DATA 10,8C,03,FC,25,F6,10,8E,03,EB
180 DATA 10,9F,72,86,21,B7,BE,C5,1C,AF
190 DATA 39
200 DATA 12,B7,FF,DF,7E,C7,06



4E20                 ********************************
4E20                 * ASSEMBLY LISTING TO TURN ON *
4E20                 * EXTRA 32K OF RAM AND MOVE   *
4E20                 * ROM AND DOS INTO RAM        *
4E20                 ********************************
4E20   4E20                    ORG    20000
4E20                           PUT    20000
4E20 8E8000          @START    LDX    #$8000
4E23 1A50                      ORCC   #$50
4E25 B7FFDE          LOOP1     STA    $FFDE
4E28 A684                      LDA    ,X
4E2A B7FFDF                    STA    $FFDF
4E2D A780                      STA    ,X+
4E2F 8CDFFF                    CMPX   #$DFFF
4E32 25F1                      BCS    LOOP1
4E34 308C1D                    LEAX   RESET,PCR
4E37 108E03EB                  LDY    #$03EB
4E3B A680            LOOP2     LDA    ,X+
4E3D A7A0                      STA    ,Y+
4E3F 108C03FC                  CMPY   #$03FC
4E43 25F6                      BCS    LOOP2
4E45 108E03EB                  LDY    #$03EB
4E49 109F72                    STY    $0072
4E4C 8621                      LDA    #$21
4E4E B7BEC5                    STA    $BEC5
4E51 1CAF                      ANDCC  #$AF
4E53 39                        RTS
4E54 12              RESET     NOP
4E55 B7FFDF                    STA    $FFDF
4E58 7EC706                    JMP    $C706
4E5B
```

modified to overcome problem number 6 and to automatically call *Toolkit* on startup.

When called the routine in **Listing one** switches the 64 into all RAM mode but still behaves as if it is in 32 mode. *Toolkit* can be stored directly into Ram using the tape previously prepared simply CLOADM "TOOLKIT". No offset is required. Do not be tempted to EXEC at this stage as you will only have to start again.

Toolkit's self destruct routine can now be disabled. **Listing two** lines 30 to 100 overwrite the destruct routine with No Operation instructions (NOP) and a final Branch Always (BRA).

Toolkit contains some command words that are identical to words used by DragonDOS. To ensure that there is no confusion some minor modifications can be made. The simple rule I have used is to change the second letter of the conflicting words in Toolkit to 'D'. Any other alternative can be made to personal choice. **Listing two** lines 120 to 220 make the following changes to command words:

AUTO becomes ADTO
ERROR becomes EDROR
BEEP becomes BDEP
ERR becomes EDR
ERL becomes EDL
FREE$ becomes FDEE$

Toolkit uses the area allocated to graphics to store pages 1 and above, DragonDOS has however moved the position of these graphics screens for its own use so there is a danger of overwriting DragonDOS. To avoid this it is necessary to add two patches to *Toolkit* that modify the commands MOVE, FRAME and CLS. These patches are inserted using **listing two**, lines 230 to 290. The first patch for FRAME and MOVE is stored between &HFA05 and &HFA0F. These patches are called by inserting two Long Branch to Subroutine commands at &HE565 and &HE556. These branches are inserted in **listing two** lines 300 to 370.

Having added the patches and made the modifications it is now possible to save all the coding to disk by SAVE"TOOLKIT.UTY",&HE000,&HFA0F,&HE002. The title 'TOOLKIT.UTY' is used later in **listing three** as the title of the program to be autorun.

The problem with a parallel printer, Dragon 64 and *Toolkit* is that *Toolkit* uses a part of Ram than a Dragon 64 looks at to determine if it is to use the serial port or the parallel port. This fools a Dragon 64 into believing it is required to send messages to the serial port when asked to output to a printer. The startup routine modifies this check (it is all now in Ram), unfortunately though this also disables the serial port.

The final **listing** number **three** is a patch for **listing one**. It allows **listing one** to be modified so that when RUN it switches to Ram mode, modifies the print routine and LOADs and RUNs the program 'TOOLKIT.UTY'. The patch is carried out by SAVEing **listing one** to disc (you must use the same line numbers as the listing), SAVE the patch to disc (make sure the line numbers are the same as **listing three**), then NEW followed by LOAD "Listing 1"

LISTING 2

```
10 REM ******************************
20 REM ***       TOOLKIT MODS      ***
30 :REM *****************************
40 :   REM ** DISSABLE SELF DESTRUCT **
50 :   REM ***************************
60 :      FOR I=0 TO 3
70 :         POKE &HE402+I,&H12
80 :      NEXT
90 :      POKE &HE40A,&H20
100 :'
110 :'
120 :    REM ***************************
130 :    REM ** CHANGE COMMAND WORDS  **
140 :    REM ***************************
150 :       POKE &HE1E5,&H44
160 :       POKE &HE201,&H44
170 :       POKE &HE228,&H44
180 :       POKE &HE25C,&H44
190 :       POKE &HE260,&H44
200 :       POKE &HE247,&H44
210 :'
220 :'
230 :   REM ***************************
240 :   REM **    ADD THE PATCHES     **
250 :   REM ***************************
260 :      FOR I=&HF9E0 TO &HFA0F
270 :         READ A$
280 :         POKE I,VAL("&H"+A$)
290 :      NEXT I
300 :      FOR I=0 TO 2
310 :         READ A$
320 :         POKE &HE565+I,VAL("&H"+A$)
330 :      NEXT I
340 :      FOR I=0 TO 2
350 :         READ A$
360 :         POKE &HE556+I,VAL("&H"+A$)
370 :      NEXT I
380 :END
390 :'
400 DATA 34,02,FE,03,FE,A6,48,27,04,8B
410 DATA 03,A7,48,A6,49,27,04,8B,03,A7
420 DATA 49,A6,4A,27,04,8B,03,A7,4A,35
430 DATA 02,30,89,FE,00,39,39
440 DATA 81,00,27,03,C3,06,00,C3,05,E0,39
450 DATA 17,14,78,17,14,AC
```

LISTING 3

```
40 :REM *** AND PATCH TO AUTO RUN TOOLKIT ***
60 :    FOR I=&H4E20 TO &H4E71
150 DATA 30,8C,26
190 DATA 86,22,30,8C,0C,9F,A6,7E,01,94
210 DATA 22,54,4F,4F,4C,4B,49,54,2E,55,54,59,
     22,00
```

followed by MERGE "Listing 2". Now SAVE the combined Basic program; this is now the startup program, and when LOADed and RUN it will automatically activate 'TOOLKIT.UTY'.

For those who want to know the operation of the machine code used in the Basic listings the assembly coding for each routine is included. It is hoped that with the aid of these short routines those with almost redundant *Toolkits* can dust them off and bring a very useful utility back to life.

```
4CF4                *******************************
4CF4                * ASSEMBLY LISTING OF PATCHES *
4CF4                * TO PREVENT MOVE, FRAME AND  *
4CF4                * CLS OVERWRITING DRAGON DOS  *
4CF4                *******************************
F9EØ   F9EØ                 ORG    $F9EØ
F9EØ                        PUT    $F9EØ
F9EØ 34Ø2           MOVE    PSHS   A
F9E2 FEØ3FE                 LDU    $Ø3FE
F9E5 A648                   LDA    8,U
F9E7 27Ø4                   BEQ    NEXT1
F9E9 8BØ3                   ADDA   #$Ø3
F9EB A748                   STA    8,U
F9ED A649           NEXT1   LDA    9,U
F9EF 27Ø4                   BEQ    NEXT2
F9F1 8BØ3                   ADDA   #$Ø3
F9F3 A749                   STA    9,U
F9F5 A64A           NEXT2   LDA    1Ø,U
F9F7 27Ø4                   BEQ    NFXT3
F9F9 8BØ3                   ADDA   #$Ø3
F9FB A74A                   STA    1Ø,U
F9FD 35Ø2           NEXT3   PULS   A
F9FF 3Ø89FEØØ               LEAX   $FEØØ,X
FAØ3 39                     RTS
FAØ4 39                     RTS
FAØ5 81ØØ           CLS     CMPA   #$ØØ
FAØ7 27Ø3                   BEQ    NEXT4
FAØ9 C3Ø6ØØ                 ADDD   #$Ø6ØØ
FAØC C3Ø5EØ         NEXT4   ADDD   #$Ø5EØ
FAØF 39                     RTS
FA1Ø
```

```
4E2Ø                   ***********************************
4E2Ø                   * ASSEMBLY LISTING TO TURN ON     *
4E2Ø                   * EXTRA 32K OF RAM, MOVE ROM      *
4E2Ø                   * AND DOS INTO RAM THEN CALL      *
4E2Ø                   * AND RUN TOOLKIT.UTY PROGRAM     *
4E2Ø                   ***********************************
4E2Ø   4E2Ø                   ORG    2ØØØØ
4E2Ø                          PUT    2ØØØØ
4E2Ø 8E8ØØØ          @START   LDX    #$8ØØØ
4E23 1A5Ø                     ORCC   #$5Ø
4E25 B7FFDE          LOOP1    STA    $FFDE
4E28 A684                     LDA    ,X
4E2A B7FFDF                   STA    $FFDF
4E2D A78Ø                     STA    ,X+
4E2F 8CDFFF                   CMPX   #$DFFF
4E32 25F1                     BCS    LOOP1
4E34 3Ø8C26                   LEAX   RESET,PCR
4E37 1Ø8EØ3EB                 LDY    #$Ø3EB
4E3B A68Ø            LOOP2    LDA    ,X+
4E3D A7AØ                     STA    ,Y+
4E3F 1Ø8CØ3FC                 CMPY   #$Ø3FC
4E43 25F6                     BCS    LOOP2
4E45 1Ø8EØ3EB                 LDY    #$Ø3EB
4E49 1Ø9F72                   STY    $ØØ72
4E4C 8621                     LDA    #$21
4E4E B7BEC5                   STA    $BEC5
4E51 1CAF                     ANDCC  #$AF
4E53 8622                     LDA    #$22
4E55 3Ø8CØC                   LEAX   TITLE,PCR
4E58 9FA6                     STX    <$A6
4E5A 7EØ194                   JMP    $Ø194
4E5D 12              RESET    NOP
4E5E B7FFDF                   STA    $FFDF
4E61 7EC7Ø6                   JMP    $C7Ø6
4E64 22544F4F4C TITLE         FCB    /"TOOLKIT.UTY"/,Ø
4E72
```

PLAYBABY

Bernice Henessey's computer answers back to the offspring

IN the early days of the Dragon, some four or five years ago, there were a great many trivial programs around, both on sale in the shops and published in the magazines. Since then the standard has improved very greatly — if you look at last year's *Dragon User* there are some very good utility programs and a number of machine code games, but very little trivia. The only problem is that there is still a requirement for a certain amount of trivia. This program seeks to redress the balance. It is aimed at the 2-4 year old toddler who annoys its older brothers and sisters by pressing a random key just as they were about to zap that last alien and go to the top of the high score table.

Basically, the program lets you press any key at random and get a response. If you are slightly more selective you get a more interesting response. When a numeric key is pressed, the corresponding number of jaguars (they should have been cats but turned out to be more menacing) is drawn on the screen and a bleep made for each one (**Figure one**). If the zero is pressed a twinkling star is displayed on the screen and *Twinkle Twinkle Little Star* is played (**Figure two**); if any other key is pressed a random sound is made and the screen cleared to a random colour with the character displayed in the middle.

The program itself is fairly trivial and needs little description as the listing is well commented and there is a flow diagram in **Figure three**. The only point of note is that the BREAK key has been disabled by the 'dirty' technique described in June 83 *Dragon User*. This works for this program but could have disastrous effects on some other programs so it is probably safer to switch OFF and ON again after running *Playbaby*.

Anyway there it is, a toddlers' introduction to computers; it will keep them quiet for at least a quarter of an hour, teach them that if you press the right key it will do what you want and also to count to 9. It's a fairly short, simple program to type in, but if you want a copy and are too lazy to type it yourself, I'll send you a copy in return for £2.25 addressed to Bernice Hennessy, 8 Towcester Rd, Blisworth, Northampton NN7 3BL.

```
10 'PLAYBABY
20 'COPYRIGHT BERNICE HENNESSY 1987
30 '*
40 CLS
50 CLEAR 500
60 '*
70 REM USE 8 GRAPHIC PAGES,FIRST 4 FORJA
GUARS ,NEXT 2 FOR STAR IN YELLOW, NEXT T
WO FOR STAR OUTLINE
80 '*
90 PCLEAR 8
100 DIM T$(42)
110 PRINT " THIS GAME CAN BE"
120 PRINT " PLAYED AND ENJOYED BY ANYONE
,"
130 PRINT " ESPECIALLY BABY"
140 PRINT " IF ZERO IS PUSHED TWINKLE "
150 PRINT " TWINKLE LITTLE STAR IS PLAYE
D."
160 PRINT " IF A NUMBER (1 TO 9) IS PUSH
ED"
170 PRINT" THAT MANY JAGUARS ARE DRAWN."
180 PRINT" IF ANY OTHER KEY IS PUSHED TH
E"
190 PRINT" CHARACTER IS PUT ON THE"
200 PRINT" SCREEN AND A SOUND MADE."
210 PRINT" THE<BREAK>KEY WILL BE DISABLE
D"
220 PRINT" SO PRESS<RESET>TO STOP."
230 PRINT
240 PRINT"PLEASE PUSH ANY KEY "
250 '*
260 REM J$ DEFINES THE SHAPE OF THE JAGU
ARS
270 '*
280 J$="C8R1F3R3E6R16E1R6G1E2D1F2G1F2D1G
2;L1H2L2G2D3F4R1D1L3H5L1D3R1D1L3;U4L4G2F
2R1D1L3H3U3G2D2R1D1L3U12;G5L3H4;"
290 '*
300 REM DEFINE TWINKLE ,TWINKLE LITTLE S
TAR
310 '*
320 DATA L4C,C,G,G,A,A,L2G
330 DATA L4F,F,E,E,D,D,L2C
340 DATA L4G,G,F,F,E,E,L2D
350 DATA L4G,G,F,F,E,E,L2D
360 DATA L4C,C,G,G,A,A,L2G
370 DATA L4F,F,E,E,D,D,L2C
380 FOR I=1 TO 42:READ T$(I):NEXT I
390 '*
400 REM DRAW STAR ON GRAPHICS    PAGES 5
&6,IN YELLOW ON BLUE AND OUTLINE ON PAGE
S 7&8
410 '*
420 PMODE 1,5
430 COLOR2,3
440 PCLS
450 ST$="BM70,70;M110,70;M128,30;M146,70
;M186,70;M156,110;M176,162;M128,120;M80,
162;M100,110;M70,70"
460 DRAW ST$
470 PAINT(80,80)
480 PMODE 1,7
490 COLOR2,3
500 PCLS
510 DRAW ST$
600 '*
610 REM DISABLE<BREAK>KEY
620 REM (THE DIRTY WAY!)
630 '*
640 POKE411,228
650 POKE412,203
660 POKE413,4
670 POKE414,237
680 POKE415,228
690 POKE410,236
1000 '*
1010 REM WAIT FOR KEY PUSH
1020 '*
1030 IN$=INKEY$
1040 IF IN$="" THEN GOTO 1030
1050 '*
1060 REM VAL(IN$)=0 IF ALPHABETIC OR ZER
O KEY PUSHED
1070 '*
1080 REM FOR ZERO KEY PUSHED GOSUB1700 T
O PLAY TWINKLE TWINKLE LITTLE STAR
1090 REM FOR OTHER NUMERIC KEY  PUSHED G
OSUB1300 TO DRAW JAGUARS
1100 REM FOR ANY OTHER CHARACTER GOSUB12
00 TO COLOUR SCREEN AND, IF POSSIBLE,PRI
NT CHARACTER
```

```
1110 '*
1120 IF IN$="0" THEN GOSUB1710 ELSE IF V
AL(IN$)=0 THEN GOSUB1210 ELSE GOSUB1310
1130 GOTO 1030
1200 '*
1210 REM SET RANDOM SCREEN COLOUR,PRINT
 CHARACTER(IF POSSIBLE) AND MAKE SOUND DE
 PENDING ON CHARACTER
1220 '*
1230 CLS RND(9)-1
1240 PRINT@208,IN$;
1250 NT=ASC(IN$)*2:SOUND NT,8
1260 RETURN
1300 '*
1310 REM NUMERIC KEY PUSHED, BUT NOT ZER
O.
1320 REM DO APPROPRIATE NUMBER OF SOUNDS
 AND DRAW JAGUARS, USING BUFF SCREEN AND
 ORANGE JAGUARS.
1330 '*
1340 PMODE 3,1
1350 COLOR8,5:PCLS
1360 SCREEN 1,1
1370 X=1:Y=10
1380 REM X,Y ARE USED TO POSITION JAGUAR
1390 N=VAL(IN$)
1400 FOR I=1 TO N
1410 SOUND 96,5
1420 X$=STR$(X):Y$=STR$(Y)
1430 '*
1440 REM DRAW JAGUAR OUTLINE
1450 '*
1460 DRAW "BM"+X$+","+Y$+"S8"+J$
1470 REM DRAW EYE
1480 DRAW "BM"+X$+","+Y$+"S8"+"BM+35,-1;
C4E1F1G1H1C8"
1490 REM PAINT JAGUAR
1500 XP=X+35:YP=Y:PAINT(XP,YP)
1510 REM POSITION FOR NEXT JAGUAR,IF ANY
1520 X=X+85:Y=Y+8:IFX>200 THEN X=1:Y=Y+3
0
1530 '*
1540 REM SLOW THINGS DOWN
1550 '*
1560 TIMER=0
1570 IF TIMER<10 THEN GOTO1570
1580 NEXT I
1590 RETURN
1700 '*
1710 REM PLAY TWINKLE  TWINKLE,  LITTLE
STAR AND DRAW STAR ON SCREEN TWINKLING Y
ELLOW ON BLUE
1720 '*
1730 PMODE 1,5
1740 REM PLAY TWO NOTES FOR EACH I:THERE
 ARE 42 NOTES IN TUNE
1750 K=0
1760 FOR I=1 TO 21
1770 FOR J=5 TO 7 STEP2
1780 PMODE1,J
1790 SCREEN 1,0
1800 K=K+1
1810 PLAYT$(K)
1820 NEXTJ,I
1830 PMODE1,5
1840 SCREEN 1,0
1850 RETURN
1860 END
```

Crossword

Please get your answers in to *Dragon User Crossword Department* by the end of the month on the front cover

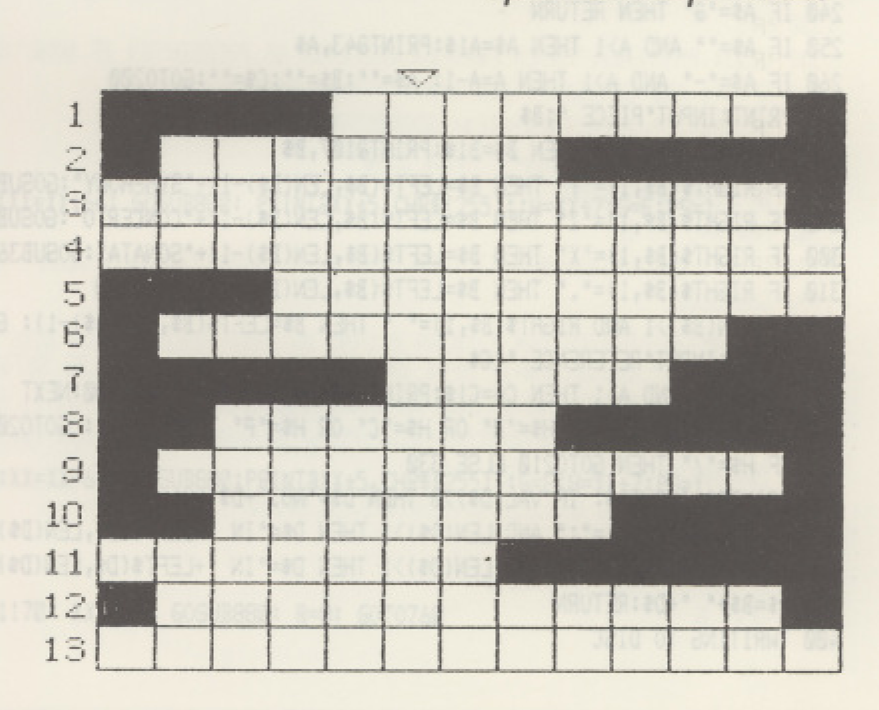
The eleventh Dragon Crossword rolls crisply out of a neat white envelope, dressed to kill, and regards ... the ninth Dragon Crossword, as it crawls damply from under a pile of slowly-rotting press releases. "Don't worry, old chap," it snaps briskly, "the Editor will get around to throwing that lot away soon. You'd better freshen up, because D O Cuin of Armagh has written to tell you that he would like a Quickbeam game, any Quickbeam game, and Patricia Hill of Surrey (an old friend of yours) is looking for an adventure of some sort. It's a good life out here, you know."

"Ha", mutters the old Crossword. "You wait till this time next month."

The phrase is HIRES GRAPHICS.

There will be a couple of free tapes from the Editor's Magic Bottomless Box for the first correct entries out of the hat each month. You can try telling us which tapes you'd like — we may have them.

All this month's answers are names of Dragon software. When the crossword is complete, the column marked with an arrow will spell out a phrase.



1 Dr. Who's a famous one, so fans cram around. (8)
2. Nail the globe — Tommy was a wizard at it.(7)
3. Air speed cars make unfriendly aliens (5,7)
4. Do call in at six to find adventure (7,6)
5. First recipe for grand prix? (7,3)
6.& 7. Pity red tiger was mad about his daily log (10,5)
8. Martial art of Grasshopper? (4,2)
9. First one raid, errant jewel thief revealed.
10. Learner in top hat knocks things over. (7)
11. Dealing is a jobbers' task, says Yosser. (3,1,3)
12. Submarine captain might say it to have a look. (2,9)
13. Are rush takers troubled by big fish riches? (5,8)

Music Catalogue

Anthony Daniels' database has records but yours could have recipes

THE value of writing your own programs is that they can be made to do exactly what you want whereas any professional software has to cater for many different needs. All my family play in orchestras and we need to keep an extensive library of music on cassettes and records as well as printed music and so we have a catalogue and quick reference system on computer. Professional databases tend to talk about defining fields and being generally obscure and abstract. My program requires none of this preliminary business. It operates in a number of short sections which will List, Edit, Store, Sort or permit Analysis, and every entry is stored in a single string which although limiting its length makes handling much easier. One can apply the techniques demonstrated to all sorts of things so I trust it will prove interesting.

The menu is stated in lines 70-170. The first task is to make up the catalogue and this is done with the command LIST. The first heading is Composer. If you press ENTER only, you will get the same name as on the previous entry and this applies throughout this section. If you wish to leave this section input '@'. To advance to the next entry press '' or to go back press either - or '!', the former will clear the previous entries, the latter will preserve them.

Next in this section comes the type of piece. I have included a shorthand here. If the right hand character of the entry is a Y it will be read as Symphony, an X will be read as Sonata and a Z as Concerto. If the word you are writing ends in one of these letters then just add an extra space after it. Having used the shorthand you move on to a further input which allows you to enter the number or key (Use D; for D major or E- for E minor). If you wish to enter both a number and key then type it in longhand. In all these entries avoid the use of commas otherwise the disc drive may read them as end of string markers.

```
5 'MUSIC CATALOGUE BY A. DANIEL 23/11/87
10 CLEAR10000
20 ERROR GOTO60: 'OPTIONAL ERROR TRAP - ONLY AVAILABLE WITH SUITABLE SOFTWARE (E.G.SUPERDOS)
30 DIME$(300),C$(300),U$(26)
40-GOSUB1660
50 A=0:GOTO70
60 CLS0: PRINT@100,"ERROR";:FOR J=1 TO 500: NEXT
70 CLS4: PRINT@99,"Read - Write - List - Edit";
80 PRINT@136,"Analysis - Sort";:PRINT@159,"    (PRESS APPROPIATE LETTER)":PRINT:PRINT"PRESS '/' TO CLEAR CURRENT FILE"
90 T$=INKEY$: IF T$="" THEN 90
100 IF T$="L" THEN GOSUB190
110 IF T$="E" THEN GOSUB740
120 IF T$="A" THEN GOTO 1240
130 IF T$="W" THEN GOSUB410
140 IF T$="/" THEN FOR F8=1 TO UB: E$(F8)="":NEXT:UB=0:A=0
150 IF T$="S" THEN GOSUB1850
160 IF T$="R" THEN GOSUB510:GOSUB1170
170 GOTO70
180 'LISTING ENTRIES
190 IF AE=1 THEN CLS:PRINT@416,E$(A): A=A-1: GOTO210
200 CLS:A1$=A$:B1$=B$:C1$=C$:D1$=D$: E$(A)=A$+"  "+B$+"  "+C$: A$="":B$="":C$="":D$="":PRINT@416,E$(A)
210 PRINT@0,A+1;:PRINT@32,"COMPOSER";:INPUTA$
220 IF A$="^" THEN A=A+1: CLS: PRINT@416,E$(A);: GOTO210
230 IF A$="!" AND A>0 THEN A=A-1: CLS: PRINT@416,E$(A);: GOTO210
240 IF A$="@" THEN RETURN
250 IF A$="" AND A>1 THEN A$=A1$:PRINT@43,A$
260 IF A$="-" AND A>1 THEN A=A-1: A$="":B$="":C$="":GOTO200
270 PRINT:INPUT"PIECE ";B$
275 IF B$="" AND A>1 THEN B$=B1$:PRINT@107,B$
280 IF RIGHT$(B$,1)="Y" THEN B$=LEFT$(B$,LEN(B$)-1)+"SYMPHONY":GOSUB360:GOTO330
290 IF RIGHT$(B$,1)="Z" THEN B$=LEFT$(B$,LEN(B$)-1)+"CONCERTO":GOSUB360:GOTO330
300 IF RIGHT$(B$,1)="X" THEN B$=LEFT$(B$,LEN(B$)-1)+"SONATA":GOSUB360:GOTO330
310 IF RIGHT$(B$,1)="." THEN B$=LEFT$(B$,LEN(B$)-1): GOTO330
320 IF LEN(B$)>1 AND RIGHT$(B$,1)=" " THEN B$=LEFT$(B$,LEN(B$)-1): GOSUB360:GOTO330
330 PRINT:INPUT"REFERENCE ";C$
335 IF C$="" AND A>1 THEN C$=C1$:PRINT@171,C$:FOR J=1 TO 1000:NEXT
340 H$=LEFT$(C$,1): IF H$="R" OR H$="C" OR H$="P" THEN A=A+1: GOTO200
350 IF H$="/" THEN GOTO210 ELSE 330
360 PRINTB$:INPUTD$: IF VAL(D$)>0 THEN D$="NO."+D$
370 IF RIGHT$(D$,1)=";" AND LEN(D$)>1 THEN D$="IN "+LEFT$(D$,LEN(D$)-1)+" MAJOR"
380 IF RIGHT$(D$,1)="-" AND LEN(D$)>1 THEN D$="IN "+LEFT$(D$,LEN(D$)-1)+" MINOR"
390 B$=B$+" "+D$:RETURN
400 'WRITING TO DISC
```

Finally enter the reference. The first letter must be C for cassette, R for record or P for printed music — or choose your own letters. Anything can follow them but you will not be able to make another entry unless one of these letters is entered. If you have made a mess up and wish to redo the entry completely enter '/'.

Having made the list, you need to WRITE it onto the disc. Of course you can enter a single file on cassette but searching for multiple files on cassette is rather a problem, and the analysis section which provides the real joy of using the program only works well from disk. The names of the files are read from data line 1720, one for each letter of the alphabet but, as you can see, I have grouped some of mine together.

The number of files depends on the size of your collection. If you put much over 100 entries per file sorting becomes rather slow. The disc syntax in the program is for SuperDOS.

To READ the files requires no special explanation so I will move on to EDIT mode. Up and down arrow keys display all the entries or pressing Q or A moves up and down the file quickly displaying only when you remove your finger. Pressing 'D' deletes and entry while 'N' adds a new space. (Press ';' and go to LIST to actually put in the details.) 'P' prints out all subsequently displayed files and 'O' turns the printer off. Pressing ':' returns to the menu at the point at which you entered it while ';' takes you back to the number you are displaying.

You do not have to put new entries in a particular slot. It is probably best to enter them first, then READ the appropriate file and then SORT. The sort routine only puts the composers' names in order but you could extend it although it might become rather slow. The routine can be seen to be working by the changing colour screens and partial printout.

Finally there is the ANALYSIS section. This is largely self explanatory from the instructions on the program. If you wish to modify this section for use with a single file on cassette may I draw your attention to the REM in line 1280. The files for analysis are read direct from disc so you should save or clear ('/') any files already on the program before starting that section.

```
410 IF UB>A THEN U=UB ELSE U=A
420 IF G$="" THEN GOSUB550
430 KILLG$:'INPUT"NAME OF FILE";G$: G$=G$+".DAT"
440 FWRITEG$;U
450 FOR E=0 TO U
460 FWRITEG$;E$(E)
470 NEXT E
480 CLOSE
490 CLS5:RETURN
500 'READING DISC
510 CLS6:PRINT"NOTE THAT TOP NUMBER IN USE IS ";
520 IF UB>A THEN PRINTUB:L=UB+1 ELSE PRINTA:UB=A:L=UB+1
530 IF L=0 THEN L=1
540 GOSUB550:GOTO600
550 PRINT"LETTER OF ALPHABET REQUIRED"
560 IF INKEY$<>"" THEN 560
570 TE$=INKEY$:IF TE$="" THEN 570
580 IF ASC(TE$)>64 AND ASC(TE$)<96 THEN G$="ML"+U$(ASC(TE$)-64)+".DAT" ELSE GOTO550
590 RETURN
600 FREADG$;U
610 FOR G=L TO U+L
620 FREADG$;E$(G)
630 NEXT G
640 CLOSE
650 UB=U+L:L=1:A=1:E1=1:E=1:RETURN
660 'SEPARATING ENTRIES FOR DISPLAY PURPOSES
670 J=INSTR(1,E$(G),"  "): IF J>0 THEN L$=LEFT$(E$(G),J)
680 K=INSTR(J+2,E$(G),"  "): IF K>0 THEN M$=MID$(E$(G),J+2,K-J):N$=RIGHT$(E$(G),LEN(E$(G))-K)
690 IF PR=2 THEN RETURN ELSE PRINT@H,L$: PRINT@H+14,N$: PRINT@H+25,M$: IF PR=1 AND PS=1 THEN 700 ELSE 720
700 GOSUB2070
710 PRINT#-2,G,L$,M$,N$
720 L$="":M$="":N$="":RETURN
730 'EDIT SECTION INCLUDING SUBROUTINES
740 CLS4: E=A: E1=A:GOSUB1170:R=0:GOTO760
750 R=0:GOSUB1020
760 '
770 IF PEEK(341)=223 THEN PRINT@XX+5,CHR$(143);:E=E+1:XX=XX+64: GOSUB880: PRINT@XX+5,CHR$(255);:H=XX+7:G=E:PS=1
    :GOSUB670:PS=0: PRINT@
 XX,E;: GOTO750
780 IF PEEK(344)=247 THEN GOSUB1050
790 IF PEEK(338)=239 THEN PR=1
800 IF PEEK(345)=247 THEN PR=0
810 IF PEEK(339)=239 THEN E=E+1:GOSUB1140:GOTO770
820 IF PEEK(339)=251 THEN E=E-1:GOSUB1140:GOTO770
830 IF PEEK(342)=223 THEN PRINT@XX+5,CHR$(143);:E=E-1:XX=XX-64: GOSUB880:PRINT@XX+5,CHR$(255);:G=E:H=XX+7:PS=1
    :GOSUB670:PS=0:PRINT@XX ,E;: GOTO750
840 IF PEEK(344)=253 THEN A=E1:AE=1: RETURN
850 IF PEEK(342)=253 THEN A=E: AE=1: RETURN
860 IF PEEK(342)=251 THEN E$(E)="":PRINT@XX,"": GOSUB1170: XX=480: GOSUB880: R=0: GOTO760
```

```
870 GOTO750
880 IF E>500 THEN E=500
890 IF XX>416 THEN XX=0: GOTO920
900 IF E<0 THEN E=0: XX=XX+64
910 IF XX<0 THEN XX=416: GOTO970 ELSE RETURN
920 CLS4
930 FOR R=0 TO 6
940 IF (E+R)>500 THEN 950 ELSE GOSUB1020
950 NEXT
960 RETURN
970 CLS4
980 FOR R=0 TO -6 STEP -1
990 IF (E+R)<0 THEN 1000 ELSE GOSUB1020
1000 NEXT
1010 RETURN
1020 IF E$(E+R)="" THEN PRINT@XX+15+R*64,"NO DATA";:PRINT@XX+6+R*64,CHR$(209); ELSE G=E+R:H=XX+7+R*64:GOSUB670
1030 PRINT@XX+R*64,E+R;
1040 RETURN
1050 IF UB<A THEN UB=A
1060 UB=UB+1: U=U+1: A=A+1
1070 FOR T=UB TO (E+1) STEP-1
1080 E$(T)=E$(T-1)
1090 NEXT
1100 E$(E)="NEW": PRINT@XX+10,E$(E): PRINT@XX,E;
1110 IF XX<416 THEN T1=XX: T2=E
1120 T2=T2+1: T1=T1+64: IF T1<448 THEN G=T2:H=T1+7:GOSUB670: PRINT@T1,T2;: GOTO1120
1130 RETURN
1140 IF E<0 THEN E=0
1150 IF E>500 THEN E=500
1160 PRINT@480,E;:RETURN
1170 B2=1: IF UB<A THEN UB=A
1180 FOR B1=1 TO UB
1190 C$=E$(B1): IF C$="" THEN 1210 ELSE E$(B2)=C$: B2=B2+1
1200 IF B1=B2-1 THEN 1210 ELSE E$(B1)=""
1210 NEXT
1220 B2=B2-1: UB=B2: RETURN
1230 'ANALYSIS SECTION
1240 FL=0:PF=0:PR=2:CLS7: PRINT@36,"ENTER EITHER COMPOSER'S NAMEOR A LETTER OF ALPHABET FOR A
     PARTICULAR FILE":PRINT@132,"TO SEARCH THE PROGRAMME IS COMPLEX IN THESE LINES BUT
     THROUGH ALL THE   FILES FOR THE CONTENTS OF AN    INDIVIDUAL TAPE OR A PARTICULAR PIECE PRESS '@' FIRST"
1250 PRINT@260,"ENTER '*' TO RETURN TO OTHERPARTS OF THE PROGRAMME": INPUT S$
1260 IF S$="@" OR S$="" THEN 1240
1270 IF LEFT$(S$,1)="@" THEN S$=RIGHT$(S$,LEN(S$)-1): FL=1 ELSE T$=LEFT$(S$,1)
1280 'VARIABLE FL AND LINE 1310 DECIDE WHETHER ALL OR ONLY 1 FILE IS TO BE SEARCHED.
1290 IF FL=1 AND LEN(S$)=1 THEN FG$=S$ ELSE FG$=""
1300 IF T$="*" THEN PR=0: GOTO70
1310 IF ASC(T$)>64 AND ASC(T$)<96 THEN G$="ML"+U$(ASC(T$)-64)+".DAT"
1320 H=1
1330 CLS2: PRINT@33,"ENTER FURTHER CRITERIA OR PRESSENTER IF NONE":INPUT S1$
1340 IF H=1 AND LEN(S$)=1 AND S1$<>"" THEN S$=S1$
1350 IF S1$<>"" THEN T$(H)=S1$: GOSUB1590: H=H+1: GOTO1330
1360 IF LEN(S$)=1 AND S1$="" THEN 1240
1370 IF FL=0 THEN 1400
1380 FOR W=1 TO 26
1390 IF W>1 AND U$(W)=U$(W-1) THEN 1500 ELSE G$="ML"+U$(W)+".DAT"
1400 GOSUB600
1410 FOR G=1 TO UB
1420 IF FG$<>"" GOSUB1610:IF PF=1 THEN GOSUB1530:GOTO1440 ELSE GOTO1490
1430 IF INSTR(1,E$(G),S$)>0 THEN PF=1:GOSUB1530 ELSE 1490
1440 IF PF=1 THEN GOSUB670:CLS6:PRINT:PRINTL$:PRINTN$:PRINTM$:PRINT:PRINTHH$: PF=0: GOTO1450 ELSE 1490
1450 I$=INKEY$:IF I$="" THEN 1450
1460 IF I$="P" THEN GOSUB2070:PRINT#-2,L$,M$,N$
1470 L$="":M$="":N$=""
1480 IF I$="/" THEN GOTO1240
1490 NEXT G
```

```
1500 IF FL=1 THEN NEXT W
1510 FL=0: FG$=""
1520 GOTO1240
1530 IF H<2 THEN RETURN
1540 HL=1
1550 IF FG$=T$(HL) THEN GOTO1570
1560 IF INSTR(1,E$(G),T$(HL))>0 THEN PF=1 ELSE PF=0
1570 IF PF=1 AND HL<H-1 THEN HL=HL+1: GOTO1550
1580 RETURN
1590 IF LEN(T$(H))=1 AND FG$="" THEN FG$=T$(H)
1600 RETURN
1610 IF INSTR(1,E$(G),S$)>0 THEN PF=1 ELSE PF=0: RETURN
1620 FJ=LEN(E$(G))
1630 FJ=FJ-1: IF FJ<1 THEN RETURN
1640 IF MID$(E$(G),FJ,2)=" "+FG$ THEN PF=1: RETURN
1650 IF MID$(E$(G),FJ,2)=" " THEN RETURN ELSE PF=0: GOTO1630
1660 FOR J=1 TO 26
1670 READ U$(J)
1680 NEXT
1690 HH$="    IF YOU WISH ANY ENTRY TO BE PRINTED OUT PRESS 'P' AFTER IT  IS DISPLAYED ON SCREEN, ELSE
     PRESS ANY OTHER KEY FOR NEXT     ENTRY. TO REDEFINE SEARCH PRESS '/'"
1700 RETURN
1710 'NAMES OF FILES, ONE FOR EACH LETTER OF THE ALPHABET
1720 DATA A-B,A-B,C-D,C-D,E-G,E-G,E-G,H-L,H-L,H-L,H-L,H-L,M-M,N-P,N-P,N-P,Q-R,Q-R,S-S,T-V,T-V,T-V,W-Z,W-Z,W-Z,W-Z
1730 'SORT ROUTINE
1740 Z=M2
1750 Z=INT(Z/2):IF Z<=0 THEN GOSUB2010:RETURN
1760 Y=M2-Z:ZZ=1
1770 X=ZZ
1780 XX=X+Z
1790 IF MID$(C$(X),P,1)<=MID$(C$(XX),P,1) THEN 1820
1800 W$=C$(X): C$(X)=C$(XX):C$(XX)=W$:PRINTC$(X):PRINTC$(XX):X=X-Z
1810 IF X>=1 THEN 1780
1820 ZZ=ZZ+1
1830 IF ZZ<=Y THEN 1770
1840 GOTO1750
1850 GOSUB1170
1860 CLS3:L=1:AB=1:S1=1:S2=UB:P=1:GOSUB1960:GOSUB1740
1870 S3=L+1: S4=L: Q=1: P=2
1880 GOSUB1930
1890 IF PQ=0 THEN S3=S2+2: S4=S2+1:GOSUB1930
1900 IF PQ=1 THEN PQ=0: P=P+1: Q=Q+1: IF P>3 THEN RETURN ELSE S3=L+1: S4=L: GOSUB1930: GOTO1890
1910 GOTO1890
1920 RETURN
1930 IF MID$(E$(S3),Q,1)=MID$(E$(S3-1),Q,1) THEN S3=S3+1 ELSE GOTO1950
1940 IF S3>UB THEN PQ=1 ELSE GOTO1930
1950 S1=S4: S2=S3-1:AB=1: JJ=RND(9)-1: CLSJJ: GOSUB1960:GOSUB1740: RETURN
1960 FOR AA=S1 TO S2
1970 C$(AB)=E$(AA): AB=AB+1
1980 NEXT
1990 M2=AB-1
2000 RETURN
2010 AB=1
2020 FOR AA=S1 TO S2
2030 E$(AA)=C$(AB):AB=AB+1
2040 NEXT
2050 RETURN
2060 'PRINTER PREPARATION - NOTE PRINTER CODES IN LINE 2070 FOR SMALLEST TYPE ON DMP115 PRINTER MAY HAVE TO BE MODIFIED
2070 IF HC=0 THEN PRINT#-2,CHR$(27);CHR$(20): HC=1
2080 IF LEN(L$)<18 THEN L$=L$+" ":GOTO2080 ELSE IF LEN(L$)>18 THEN L$=LEFT$(L$,18)
2090 IF LEN(M$)<55 THEN M$=M$+" ":GOTO2090 ELSE IF LEN(M$)>55 THEN M$=LEFT$(M$,55)
2100 IF LEFT$(N$,1)=" " THEN N$=RIGHT$(N$,LEN(N$)-1):GOTO2100
2110 RETURN
```

Phoneticode

***J F Rowles** second-guesses spellings for sorting.*

THE successful operation of data files for say names and addresses, books and authors, record collections etc. is very dependent on the sort-keys used for retrieval of data. In most cases the surname is used as the primary key for retrieval. While this operates very successfully, instances do occur where uncertainty of the exact spelling can result in repeated attempts of all known variants or a range selection being made. As an example, consider the variations found for 'Smith'. A perusal of my local telephone directory revealed the following possibilities: Smith, Smeeth, Smit, Smidt, Smyth and Smythe. On a simple data base this would probably involve a separate search for each of the variants (assuming you are aware of all the possibilities) or a range selection of say SMAAA to SMZZZ, which of course will select all data commencing with the letters 'SM', which on a large data base could be quite extensive.

A method much used by professional data base enquiry systems is to interrogate the files for phonetically similar names where the exact spelling is unknown, or alternatively to check initially for the believed correct spelling and if no match is found then resort to phonetics. This has much to commend it in that you may be making an enquiry using the correct spelling but the original data was entered incorrectly.

While this may seem a daunting prospect to expect the simple home micro to perform such a task, it is in fact fairly easy to achieve. The system detailed below is modelled on one of the systems used in the professional data handling world. It should be realised that phonetics are language controlled (spoken that is, not computer), and any phonetic encoding will only work on the language for which it was designed and to a lesser degree on similar languages and not at all on others.

Successful phonetic encoding merely requires the grouping of like-sounding letters together, as follows:

1. B,F,P,V
2. C,G,J,K,Q,S,X,Z
3. D,T
4. L
5. M,N
6. R

Try saying them phonetically, as a child does (or used to in my day) when first learning the alphabet. See the similarities?

The more astute will have noticed that all the vowels together with Y, H and W are missing from the above groups. These are totally unnecessary for phonetic encoding and are ignored unless they are the first letter in the word or name. Try it yourself and see. Pick any word at random, write it down, pronounce it out loud, then rewrite the word omitting these letters and attempt to pronounce it. Unless you are very unlucky the second word should be recognisable to the ear. This is the basis of phonetic encoding.

Now to practicalities. The code is assembled by retaining the first letter of the name or word to be encoded as the first character of the code. Subsequent letters are tested for consecutive duplication, only the first occurrance being retained, and these letters are then assigned a numeric character according to which of the phonetic groups they belong. The whole code is then retested for consecutive duplication and truncated or expanded by the addition of trailing zeros to four characters long. This then forms the phonetic code of that name or word.

This may sound complex, so consider the following example:

Name to be encoded = 'SMITH'

Following the rules above the first character of the code will be the first letter of the name ie S. The second letter is not a duplicate of the first so use the look up chart above, M falls in group 5. This is the second character of the code. Likewise for the remaining letters, I is ignored, T is in group 3 and H is ignored. The code is therefore 'S53'. This is expanded to four characters by the addition of trailing zeros, so the final phonetic code for 'SMITH' is 'S530'. Try this for the other variants on the name 'SMITH' mentioned previously, you will find that they all encode to 'S530'. So it works for 'SMITH'. How about other names? Try a few you can think of — you should be pleasantly surprised. Of course there are a few names that will defy these methods but these are usually of the more exotic or historical species. (Try 'CHALMONDLEY' which is pronounced 'CHUMLEY' — it does not produce a phonetic code which is compatible with its orthographics). However for the more common names and some unusual variants on spelling the encoding works well (Try 'MAINWAIRING' and 'MANNERING' — the phonetic codes are identical).

Now to the programs themselves. The main listing for the coding has been written as a sub-routine and in its existing form is ready to append to any program you may wish. The word or name to be encoded is IN$ and the resulting Phoneticode is CODE$. If you plan on using this system to any great degree it would be worth considering adding the phonetic codes of your principal sort items to your main data base to speed selection. The second listing is merely a short program to append to the main listing so that you can experiment with different words and names to find the results.

If you operate a large data base you may find that truncation of the final code to four characters results in too many selections. The cure is simply to enlarge the size of the code to five or six characters long by making the appropriate alterations to lines 9015, 9016 and 9017.

The program has been kept simple deliberately to aid transportability between different machines (rumour has it that there are other machines than the Dragon, but as yet the author is not fully convinced.)

No doubt the machine-code experts among you will re-write the Basic program, but the object of the article was more to provoke thought and experimentation than to present a ready-made machine-code routine. The addition of one of the usual 'speed-up' pokes in the Basic program if your Dragon is suitable will be of great benefit to those planning to use it as it stands.

```
9000 '***************************
9001 '*** PHONETIC ENCODING ***
9002 '***************************
9003 '*** WRITTEN JULY 1986 ***
9004 '***    PROGRAM ID      ***
9005 '***   PHONETIC.BAS     ***
9006 '***  C.  J.F.ROWLES    ***
9007 '***************************
9008 CHECK$=LEFT$(IN$,1)
9009 IF MID$(IN$,2,1)=CHECK$ THEN Y=3 ELSE Y=2
9010 FOR X=Y TO LEN(IN$)
9011 C$=MID$(IN$,X,1)
9012 GOSUB 9019
9013 NEXT
9014 GOSUB 9027
9015 IF LEN(CODE$)>4 THEN CODE$=LEFT$(CODE$,4)
9016 IF LEN(CODE$)<4 THEN CODE$=CODE$+"0"
9017 IF LEN(CODE$)<4 THEN 9016
9018 RETURN
9019 IF C$="B"OR C$="F"OR C$="P" R C$="V"THEN CHECK$=CHECK$+"1"ELSE 9020:RETURN
9020 IF C$="C"OR C$="G"OR C$="J"OR C$="K"OR C$="Q"OR C$="S"OR C$="X"OR C$="Z"
     THEN CHECK$=CHECK$+"2" ELSE 9022:RETURN
9022 IF C$="D" OR C$="T" THEN CHECK$=CHECK$+"3" ELSE 9023:RETURN
9023 IF C$="L" THEN CHECK$=CHECK$+"4" ELSE 9024:RETURN
9024 IF C$="M" OR C$="N" THEN CHECK$=CHECK$+"5" ELSE 9025:RETURN
9025 IF C$="R" THEN CHECK$=CHECK$+"6"
9026 RETURN
9027 CODE$=LEFT$(CHECK$,2)
9028 FOR X=3TO LEN(CHECK$)
9029 Z$=MID$(CHECK$,X,1)
9030 Y$=MID$(CHECK$,(X-1),1)
9031 IF Z$=Y$ THEN NEXT ELSE CODE$=CODE$+Z$ :NEXT
9032 RETURN
```

```
10 CLS
20 INPUT"WORD TO ENCODE";IN$
30 GOSUB 9000
40 PRINTIN$,CODE$
50 GOTO 20
```

Expert's Arcade Arena

Write to 'The Expert' at Dragon User
49 Alexandra Road
Hounslow, Middlesex, TW3 4HP

HELLO PEEPS, and welcome to the second games round-up, which feature almost all of the arcade games which are still readily available from third party sources, and were not included in the original round up.

The first dozen are available from Computape (don't forget the new address), *Tim Love's Cricket* from John Penn, and the last five are available from Preston's.

The original format has been used again, with all marks out of five, with three being the ideal speed. The overall rating is an indicator of how necessary the program is to a hardened games player.

As for categories, well, shoot-em-up games require an itchy trigger finger, collection games are for *Manic Miner* hunchback clones, strategy games require a bit more thought than Da Average Game, Boo-Boo, boy, and as for adventure and sport games, well, you can work that out for yourself.

There's no room for pokes this time, but you can look forward to the *Superkid* cheat codes and a few more, so now my time is up, I'll bug off, innit.

Title	Graphics	Speed	Type	Comment	Rating
Airball	5	3	Adventure	Ed Scio's conversion is now a big hit on the Atari as well.	5
Space Wreck	3	2	Shoot'em up	A mixture of Battlezone and Asteroids which comes a poor second to Rommel-3D.	1
Tanglewood	4	NA	Adventure	Not being strictly an arcade game, this will have to receive an average rating, although it is very popular.	3
Junior's Revenge	3	3	Collection	The King 2 — only Mario has changed his name to Luigi, and is now the baddie.	2
Time Bandit	3	4	Adventure	One of the first Dragon arcade advantures, which has stood the test of time.	4
Cuthbert and the Golden Chalice	2	3	Collection	A simple but quite enjoyable obstacle course game.	2
Pinball	2	2	Strategy	An unbelievably easy game which is unfortunately the only one of its kind for the Dragon.	0
Fire Force	4	4	Shoot'em up	What more can I say.	1
Indoor Football	5	3	Sport	Without a doubt the best football game on the Dragon.	5
Superkid	5	3	Collection	A faithful copy of the arcade classic onderboy, which was Wayne Smithson's best.	5
Screaming Abdabs	4	3	Collection	The hardest Manic Miner clone that I've played.	3
Crazy Foota 2 and 3	2	1-5	Sport	Great fun, and especially good in the two players mode.	3
Tim Love's Cricket	5	1	Sport	Totally realistic cricket game which is as boring as the real thing.	2
Rola-ball	4	4	Adventure	Buy it now that the bug has been sorted out.	5
Boulder Crash	4	3	Collection	Better than Microdeal's Stone Raider II, with the added advantage of extra screens from Paul Burgin.	4
The Bells	1	5	Collection	A hunchback type game which, due to its sheet speed, is impossible to play with a joystick.	1
Bugdiver	2	3	N/A	A mediocre Frogger clone which is the only one of its type still available from the several which were made.	3
Vegas Jackpot	2	NA	N/A	This game was co-written many years ago by Jason Orbaum. Good but a bit easier than the money-gobbling machines.	2

Polar graphics

K. Redhead finds a formula for entertainment

THIS program has been adapted for the Dragon from an original listing in a book written by Czes Kosniowski called *Fun Mathematics on your Microcomputer*, but greatly expanded to provide additional facilities such as disc storage and retrieval of graphs, hard copy, etc.

I have classified it as being primarily educational insofar as it illustrates the translation of simple mathematical functions into a visual medium, and as such it could possibly be of interest to schools which still use Dragon computers. On the other hand, it is equally entertaining to discover the almost endless variety of patterns capable of being generated by these formulae.

Once the overall structure of the program is understood you will see that it can easily be converted to illustrate other mathematical functions — quadratic equations, for example — or indeed for any subject where formulae can be expressed in graphical terms. There is ample scope for experiment.

The program is written for disk operation; no cassette version is available. SAVE and RUN 'POLAR'

Detailed notes on the program's operation have not been considered necessary as it is fairly straightforward and REM statements are included where appropriate. However, the following points may be of interest:

Lines 80-140 Screen Inverter, giving green on black printing. First published by Brian Cadge in *Popular Computing Weekly*, July, 1983. If you prefer orange text, change line 140 to POKE 32644,13. The routine will work on a 32K machine and on a Dragon 64 running in 32K mode.

Lines 170-1060 Draw String$ for hi-res character set, giving 51 CPL when used at normal size (SC=4). Each character has been created on a 4 x 9 grid. If you already have a similar character set on file you

```
10 REM PROGRAM: POLAR GRAPHICS
20 REM CLASSIFICATION: EDUCATION
30 REM COMPUTER: DRAGON 32
40 REM ORIGINAL AUTHOR: C.KOSNIOWSKI. 1983
50 REM DRAGON VERSION BY: K.REDHEAD. 1988
60 REM
70 REM SCREEN INVERTER
80 PCLEAR 8:CLEAR 300,32580
90 FOR I=1 TO 184:READA$:Z=VAL("&H"+A$):CS=CS+Z:POKE I+32580,Z:NEXT
100 DATA 8E,7F,63,BF,1,68,8E,7F,DE,BF,1,6B,8E,7F,F6,BF,1,A1,86,7E,B7,1,67,B7,1,6
A,B7,1,A0,39,7D,0,6F,27,1,39,32,62,34,16,8E,FF,C0,A7,84,A7,2,A7,4,A7,6,A7,A,A7,C
,A7,E,A7,88,10,A7,9,86,D,B7,FF,22,35,2,34,2,BE,0,88,81,8,26,8,86
110 DATA 20,A7,84,A7,82,20,1C,81,D,26,4,8D,32,20,14,81,80,24,E,81,20,25,C,81,60,
24,4,84,BF,20,2,80,20,A7,80,BF,0,88,8C,5,FF,23,12,8E,4,0,EC,88,20,ED,81,8C,5,E0,
25,F6,BF,0,88,8D
120 DATA 2,35,96,86,20,A7,80,1F,10,C4,1F,26,F6,39,81,C,27,1,39,34,12,86,20,8E,4,
0,BF,0,88,A7,80,8C,6,0,25,F9,35,92,27,1,39,32,62,20,E6
130 IF CS<>17105 THEN PRINT"DATA ERROR":SOUND 1,2:STOP
140 EXEC 32581:POKE 32644,5
150 REM HI-RES CHARACTER SET (51 CPL)
160 DIM A$(90)
170 A$(0)="BR3"
180 A$(1)="BR1D4BD2D0"
190 A$(2)="BR1D2BR2U2"
200 A$(3)="BR1ND2BR1D2NR1L2BD2R1ND2R1ND2R1"
210 A$(4)="BR2D1NR1L1G1F1R1F1G1ND1L2"
220 A$(5)="R1BR2D1G3D1BR2R1"
230 A$(6)="BR3BD6L2H1U1E3H1L1G1D1F3D1"
240 A$(7)="BR2ND1R1D1NL1D1G1"
250 A$(8)="BR2G2D2F2"
260 A$(9)="F2D2G2"
270 A$(10)="BD1D0F1ND1R1NE1D1NR1L2BR1D1NG1R1NU1F1"
280 A$(11)="BR1BD2D2BH1R2"
290 A$(12)="BR1BD5ND1R1D1NL1D1G1"
300 A$(13)="BD3R3"
310 A$(14)="BR1BD5D1R1U1L1"
320 A$(15)="BR3BD1D1G3D1"
330 A$(16)="NR3D6R3U6"
340 A$(17)="BD1E1D6NL1R1"
350 A$(18)="BD1E1R1F1D1G3D1R3"
360 A$(19)="R2F1D1G1NL1F1D1G1L2"
370 A$(20)="D6R3BU2BL1D3"
380 A$(21)="NR3D3R2F1D1G1L2"
390 A$(22)="BR1G1D4F1R1E1U1H1L2"
400 A$(23)="R3D2G3D2"
410 A$(24)="BR2L1G1D1F1R1F1D1G1L1H1U1E1R1E1U1H1"
420 A$(25)="BR2NF1L1G1D1F1R2NU2D2G1L1"
```

```
430 A$(26)="BD2R1D1L1NU1BD2R1D1L1U1"
440 A$(27)="BD2R1D1L1NU1BD2ND1R1D1NL1D1G1"
450 A$(28)="BR2BD1G2F2"
460 A$(29)="BD2NR3BD2NR3"
470 A$(30)="BF1F2G2"
480 A$(31)="BD1E1R1F1D1G1D1BD2D0"
490 A$(32)="BD1E1R1F1D5L2H1U1E1R2"
500 A$(33)="BR2L1G1D5BU3R3ND3U2H1"
510 A$(34)="ND6R2F1D1G1NL2F1D1G1L2"
520 A$(35)="BR3BD1H1L1G1D4F1R1E1"
530 A$(36)="ND6R2F1D4G1L2"
540 A$(37)="NR3D3NR2D3R3"
550 A$(38)="NR3D3NR2D3"
560 A$(39)="BR3BDH1L1G1D4F1R2U2L1"
570 A$(40)="D6BU3R3ND3U3"
580 A$(41)="R2BL1D6NL1R1"
590 A$(42)="R3BL1D5G1H1"
600 A$(43)="ND6BR3D1G2NL1F2D1"
610 A$(44)="D6R3"
620 A$(45)="ND6F1D1R1U1E1D6"
630 A$(46)="ND6F3NU3D3"
640 A$(47)="BR2L1G1D4F1R1E1U4H1"
650 A$(48)="ND6R2F1D1G1L1"
660 A$(49)="BR2L1G1D4F1R2NH1NF1U5H1"
670 A$(50)="ND6R2F1D1G1NL2F1D2"
680 A$(51)="BR3BD1H1L1G1D1F1R1F1D1G1L1H1"
690 A$(52)="R2BL1D6"
700 A$(53)="D5F1R1E1U5"
710 A$(54)="D4F2E1U5"
720 A$(55)="D6E1U1R1D1F1U6"
730 A$(56)="D2F1R1F1D2BL3U2E1BR1E1U2"
740 A$(57)="D2F1R2NU3D2G1L2"
750 A$(58)="R3D1G3D2R3"
760 A$(59)="NR2D6R2"
770 A$(60)="D1F3D1"
780 A$(61)="R2D6L2"
790 A$(62)="BD2NR3BD2NR3BD2NR3":REM
    A$(62)="BR1D1G1D1BR2U1H1"
800 A$(63)="BD8R3"
810 A$(65)="BD2BR1R1F1D3L2H1E1R2"
820 A$(66)="D6R2E1U2H1L2"
830 A$(67)="BR3BD2L2G1D2F1R2"
840 A$(68)="BR3D6L2H1U2E1R2"
850 A$(69)="BD4R2E1H1L1G1D2F1R2"
860 A$(70)="BR3L1G1D5BU3NL1R2"
870 A$(71)="BR2BD2NF1L1G1D2F1R2NU3D1G1L2"
880 A$(72)="D6BU4R2F1D3"
890 A$(73)="BR1D0BD2D4NL1R1"
900 A$(74)="BR3D0BD2D4G1L1H1"
910 A$(75)="D6BU2R1NE2F2"
920 A$(76)="R1D6NL1R1"
930 A$(77)="BD2ND4F1D1R1U1E1D4"
940 A$(78)="BD2ND4R2F1D3"
950 A$(79)="BR2BD2L1G1D2F1R1E1U2H1"
960 A$(80)="BD2ND6R2F1D2G1L2"
970 A$(81)="BR3BD2L2G1D2F1R2ND2U4"
980 A$(82)="BD2D4BU2E2R1"
990 A$(83)="BR3BD2L2G1F1R1F1G1L2"
1000 A$(84)="BR1D6R1E1BH3R2"
1010 A$(85)="BD2D3F1R1E1U3"
1020 A$(86)="BD2D2F2E1U3"
1030 A$(87)="BD2D4E1U1R1D1F1U4"
1040 A$(88)="BD2D1F1R1F1D1BL3U1E1BR1E1U1"
1050 A$(89)="BD2D3F1R2NU4D1G1L2"
1060 A$(90)="BD2R3D1G1L1G1D1R3"
1070 GOTO 1260
1080 REM SCALING SUBROUTINE FOR 51 CPL TEXT
1090 XS=(SC+(SC/4)):YS=(SC*2.5):RETURN
1100 REM LINE CENTERING ROUTINE
1110 LN=LEN(M$)*XS:X=INT((256-LN)/2):RETURN
1120 REM UNDERLINING ROUTINE
1130 LN=LEN(M$)*XS:LINE((X-LN),(Y+(YS-2)))-
     ((X-(SC/2)),(Y+(YS-2))),PSET:RETURN
1140 REM HI-RES PRINT ROUTINE
1150 FOR I=1 TO LEN(M$)
1160 CH$=A$(ASC(MID$(M$,I,1))-32)
1170 GOSUB 1220
1180 X=X+XS
1190 IF X+XS>255 THEN X=0:Y=Y+YS
1200 NEXT I
1210 RETURN
```

```
1220 X$=STR$(X):Y$=STR$(Y):SC$=STR$(SC):CO$=STR$(CO):AN$=STR$(AN)
1230 DRAW"BM"+X$+","+Y$+"S"+SC$+"C"+CO$+"A"+AN$+CH$
1240 RETURN
1250 REM MAIN PROGRAM BEGINS
1260 DIM FM$(16)
1270 SX=256:SY=180:RA=0.92:PI=4*ATN(1)
1280 HX=SX/2:HY=SY/2:S$=CHR$(42):SP$=STRING$(3,32)
1290 FOR I=1 TO 16:READ FM$(I):NEXT
1300 REM CREDITS AND PROGRAM INFORMATION
1310 CLS
```

could substitute it and save a fair bit of typing, provided that two points are born in mind:

1) The starting point for drawing a character is the top-left corner of the grid. If your own character set is constructed differently, you will need to take this into account and alter the settings of the 'Y' coordinates in order to place the text in the correct position on the screen.

2) Line 790 redefines the up-arrow key to produce the mathematical symbol 'p' (See line 3640). The normal draw string for this character is given in the accompanying REM statement.

Line 1090 For the purpose of printing text on the hi-res screen, this line defines the steps between the starting points of the individual characters (XS) and between lines of text (YS) as functions of the scale (SC) being used, so that altering SC automatically adjusts XS and YS accordingly. As written, the parameters are set for 51 CPL printing at normal size. To use the routing with other character sets of different sizes, make the following alterations: for 42 CPL printing, ie with each character's width occupying five pixels, make XS=(SC*1.5) For 64 CPL printing, ie with each character's width occupying three pixels, make XS=SC The YS parameter would probably not require altering, but this would depend on the number of text lines per screen.

Line 1110 This subroutine automatically centres a line of text on the hi-res screen.

Line 1130 This subroutine automatically underlines a hi-res text string. However, it cannot be used if the text to be underlined continues over from one line onto the next, nor, for that matter, if the text concerned reaches the end of a line, thus causing the print position to move to the beginning of the next line. There must be a way to get round this, but I haven't managed to figure it out yet.

Lines 1150-1240 This is the routine for printing on the hi-res screen. I've used this method ever since Iolo ap Gwynn demonstrated its versatility in the July 1984 edition of *Dragon User* - it's one of several methods of putting text on the hi-res screen that have appeared over the years. I like it because of its simplicity and ease of use.

Line 1270 Among other things, this line limits the depth of the hi-res screen to 180 pixels in order to reserve a line for labeling the graphs.

Lines 1760-2010 Sixteen formulae are contained in the program, displayed in two groups by these lines. It would be fairly easy to add more if required, increasing the number of pages as necessary and

continued on page 25

```
1320 PRINTTAB(4)STRING$(23,S$)
1330 PRINTTAB(4)S$+"DRAGON POLAR GRAPHICS"+S$
1340 PRINTTAB(4)S$ TAB(7)"BY K.REDHEAD 1988" TAB(26)S$
1350 PRINTTAB(4)S$ TAB(8)"ADAPTED FROM AN" TAB(26)S$
1360 PRINTTAB(4)S$ TAB(6)"ORIGINAL PROGRAM BY" TAB(26)S$
1370 PRINTTAB(4)S$+"CZES KOSNIOWSKI  1983"+S$
1380 PRINTTAB(4)STRING$(23,S$)
1390 PRINT
1400 PRINT"THIS PROGRAM ENABLES YOU TO PLOT";
1410 PRINT"THE GRAPHS OF POLAR FUNCTIONS. A";
1420 PRINT"LIBRARY OF FUNCTIONS IS PROVIDED";
1430 PRINT"TO ENABLE YOU TO EXPERIMENT WITH";
1440 PRINT"DIFFERENT VALUES."
1450 PRINT
1460 PRINTTAB(1)"PRESS THE SPACEBAR TO CONTINUE";
1470 K$=INKEY$:IF K$="" OR K$<>CHR$(32) THEN 1470
1480 REM DISPLAY HELP SCREEN
1490 GOSUB 1500:GOTO 1640
1500 CLS
1510 PRINTTAB(3)"POLAR GRAPHICS - HELP SCREEN"
1520 PRINTTAB(3)STRING$(28,45):PRINT
1530 PRINT"FACILITIES AVAILABLE:-":PRINT
1540 PRINTTAB(4)"F>FUNCTIONS LIBRARY"
1550 PRINTTAB(4)"S>SAVE GRAPH TO DISK"
1560 PRINTTAB(4)"L>LOAD GRAPH FROM DISK"
1570 PRINTTAB(4)"P>PRINT GRAPH"
1580 PRINTTAB(4)"D>VIEW DIRECTORY"
1590 PRINTTAB(4)"H>RETURN TO HELP SCREEN"
1600 PRINTTAB(4)"T>TUTORIAL"
1610 PRINTTAB(4)"E>TERMINATE PROGRAM"
1620 PRINT:PRINT"WHICH DO YOU REQUIRE?"
1630 RETURN
1640 K$=INKEY$:IF K$=""THEN 1640
1650 P=INSTR("FSLPDHTE",K$):IF P>0 THEN 1680
1660 PRINTTAB(3)"INVALID - PLEASE RE-ENTER"
1670 WAIT 2000:GOTO 1490
1680 ON P GOSUB 1770,2500,2660,2910,3250,1500,3370,1700
1690 IF P<8 THEN 1640 ELSE END
1700 CLS
1710 PRINT@137,"POLAR GRAPHICS"
1720 PRINT@169,STRING$(14,45)
1730 PRINT@263,"PROGRAM TERMINATED"
1740 PRINT@295,STRING$(18,45)
1750 RETURN
1760 REM DISPLAY FUNCTIONS LIBRARY
1770 CLS
1780 PRINTTAB(4)"FUNCTIONS LIBRARY PAGE 1"
1790 PRINTTAB(4)STRING$(24,45):PRINT
1800 FOR I=1 TO 8
1810 PRINTTAB(6)CHR$(I+64);CHR$(62);"R=";FM$(I)
1820 NEXT
1830 PRINTTAB(6)CHR$(90);CHR$(62);"VIEW NEXT PAGE"
1840 PRINT:PRINT"WHICH DO YOU REQUIRE?"
1850 K$=INKEY$:IF K$=""THEN 1850 ELSE N=ASC(K$)-64
1860 IF N=26 THEN 1890 ELSE IF N>=1 AND N=<8 THEN 2010
1870 PRINTTAB(3)"INVALID - PLEASE RE-ENTER"
1880 WAIT 2000:GOTO 1770
1890 CLS
1900 PRINTTAB(4)"FUNCTIONS LIBRARY PAGE 2"
1910 PRINTTAB(4)STRING$(24,45):PRINT
1920 FOR I=9 TO 16
1930 PRINTTAB(6)CHR$(I+64);CHR$(62);"R=";FM$(I)
1940 NEXT
1950 PRINTTAB(6)CHR$(90);CHR$(62);"VIEW PREVIOUS PAGE"
1960 PRINT:PRINT"WHICH DO YOU REQUIRE?"
1970 K$=INKEY$:IF K$=""THEN 1970 ELSE N=ASC(K$)-64
1980 IF N=26 THEN 1770 ELSE IF N>=9 AND N=<16 THEN 2010
1990 PRINTTAB(3)"INVALID - PLEASE RE-ENTER"
2000 WAIT 2000:GOTO 1890
2010 ON N GOSUB 4120,4140,4160,4180,4200,4220,4240,4260,
     4280,4300,4320,4340,4360,4380,4400,4420
2020 CLS
2030 PRINT"FORMULA CHOSEN IS:"
2040 PRINT "R=";FM$(N):PRINT
2050 PRINT"FOR STANDARD PLOT USE A=1,B=1"
2060 PRINT:INPUT"ENTER VALUE FOR A ";A
2070 INPUT"ENTER VALUE FOR B ";B
2080 REM CALCULATE RANGE OF R
2090 CLS
2100 PRINT@201,"CALCULATING"
2110 PRINT@233,STRING$(11,45)
2120 M=1.0^-30
2130 FOR Z=0 TO 2*PI STEP 0.1
2140 R=ABS(FNA(Z)):IF M<R THEN M=R+0.1
2150 NEXT
2160 CLS
2170 PRINT"AN IDENTITY NUMBER IS REQUIRED"
2180 PRINT"FOR THIS GRAPH. DO YOU WISH TO"
2190 PRINT"CHECK THE DIRECTORY BEFORE YOU"
2200 PRINT"ALLOCATE ONE (Y/N)?"
2210 K$=INKEY$:IF K$=""THEN 2210
2220 IF K$<>"Y" AND K$<>"N" THEN 2210 ELSE IF K$="Y"
     THEN GOSUB 3250
2230 PRINT:INPUT"ENTER GRAPH IDENTITY NUMBER ";ID
2240 ID$=STR$(ID):ID$=MID$(ID$,2):A$=STR$(A):A$=MID$
     (A$,2):B$=STR$(B):B$=MID$(B$,2)
2250 PRINT:PRINT"A NOTE WILL SOUND WHEN THE GRAPH";
2260 PRINT"IS COMPLETE. YOU MUST THEN PRESS";
2270 PRINT"THE REQUIRED KEY (fslpdhte) WHEN";
2280 PRINT"YOU WISH TO PROCEDE."
2290 PRINT@481,"PRESS THE SPACEBAR TO CONTINUE";
2300 K$=INKEY$:IF K$=""OR K$<>CHR$(32) THEN 2300
2310 REM LABEL AND PLOT GRAPH
2320 PMODE 4,1:SCREEN 1,0:PCLS
2330 M$="GRAPH No."+ID$+SP$+"R="+FM$(N)+SP$+"A="+A$+SP$+
     "B="+B$
2340 Y=183:SC=4:GOSUB 1090:AN=0:CO=1
2350 GOSUB 1110:GOSUB 1150
2360 FOR Z=0 TO 2*PI STEP 0.01
2370 R=FNA(Z)
2380 U=HX+HY*RA*COS(A*Z)*R/M
2390 IF U<0 OR U>SX THEN 2450
2400 V=HY+HY*SIN(B*Z)*R/M
2410 IF V<0 OR V>SY THEN 2450
2420 PSET(U,SY-V):IF Z=0 THEN 2440
2430 LINE(X,Y)-(U,SY-V),PSET
2440 X=U:Y=SY-V
2450 NEXT
2460 IF A=1 AND B=1 THEN PSET(0,181)
2470 SOUND 100,80
2480 RETURN
2490 REM SAVE GRAPH TO DISK
2500 CLS
2510 PRINTTAB(10)"SAVE GRAPH"
2520 PRINTTAB(10)STRING$(10,45):PRINT
2530 GOSUB 2540:GOTO 2580
2540 PRINT"CHECK THAT THE CORRECT DISK IS"
2550 PRINT"READY IN THE DRIVE & PRESS THE"
2560 PRINT"SPACEBAR TO CONFIRM."
2570 K$=INKEY$:IF K$=""OR K$<>CHR$(32) THEN 2570 ELSE RETURN
2580 F$="GRAPH"+ID$+".HRG"
2590 SAVE F$,3072,9215,6144
2600 M$=F$+" SAVED":LN=LEN(M$)
2610 PRINT:PRINT:PRINTTAB(16-LN/2)M$
2620 PRINTTAB(16-LN/2)STRING$(LN,45)
2630 PRINT@450,"PRESS REQUIRED KEY (fslpdhte)"
2640 RETURN
2650 REM LOAD GRAPH FROM DISK
2660 CLS
2670 PRINTTAB(10)"LOAD GRAPH"
2680 PRINTTAB(10)STRING$(10,45):PRINT
2690 GOSUB 2540
2700 PRINT:PRINT"GRAPHS ARE FILED BY IDENTITY"
2710 PRINT"NUMBERS.  DO YOU WISH TO CHECK"
2720 PRINT"THE DIRECTORY BEFORE SELECTING"
2730 PRINT"ONE (Y/N)?"
2740 K$=INKEY$:IF K$=""THEN 2740
2750 IF K$<>"Y" AND K$<>"N" THEN 2740 ELSE IF K$="Y"
     THEN GOSUB 3290
2760 CLS
2770 PRINT"ENTER THE IDENTITY NO. OF THE"
2780 PRINT"GRAPH YOU WISH TO VIEW";
2790 INPUT ID:ID$=STR$(ID):ID$=MID$(ID$,2)
2800 F$="GRAPH"+ID$+".HRG"
2810 PRINT:PRINT"AFTER THE GRAPH HAS BEEN LOADED"
2820 PRINT"& DISPLAYED, PRESS THE REQUIRED"
2830 PRINT"KEY (fslpdht) WHEN YOU WISH TO"
2840 PRINT"PROCEDE."
2850 PRINT@481,"PRESS THE SPACEBAR TO CONTINUE";
2860 K$=INKEY$:IF K$=""OR K$<>CHR$(32) THEN 2860
2870 PMODE 4,1:SCREEN 1,0:PCLS
2880 LOAD F$
2890 RETURN
2900 REM PRINT GRAPH
2910 CLS
2920 PRINTTAB(10)"PRINT GRAPH"
2930 PRINTTAB(10)STRING$(11,45):PRINT
2940 PRINT"PREPARE PRINTER & PRESS ANY KEY"
2950 PRINT"WHEN READY TO PROCEDE."
2960 K$=INKEY$:IF K$=""THEN 2960
2970 PRINT:PRINT"PRINTING WILL TAKE A FEW MINUTES";
2980 PRINT:PRINTTAB(10)"PLEASE WAIT"
2990 PRINTTAB(10)STRING$(11,45)
3000 PRINT £-2,TAB(25)"";
3010 PRINT £-2,CHR$(27);"-";CHR$(1);CHR$(14);CHR$(27);"E";
3020 PRINT £-2,"POLAR GRAPHICS"
3030 PRINT £-2,CHR$(13);CHR$(10);CHR$(14);CHR$(27);"F";
3040 A=PPOINT(0,18):IF A=1 THEN T$="(STANDARD)" ELSE T$="
     (ENHANCED)"
3050 PRINT £-2,TAB(4)"PLOT OF POLAR FUNCTION "+T$
3060 PRINT £-2,CHR$(27);"-";CHR$(0);CHR$(20);
3070 PRINT £-2
3080 REM HI-RES SCREEN DUMP (FOR BROTHER HR-5 PRINTER)
3090 PRINT £-2,CHR$(27);"1";CHR$(27);"M";
3100 PRINT £-2:Y=0
3110 PRINT £-2,CHR$(9);CHR$(9);CHR$(9);
3120 PRINT £-2,CHR$(27);"K";CHR$(255);CHR$(0);
3130 FOR X=0 TO 255
```

```
3140 A=PPOINT(X,Y)*128+PPOINT(X,Y+1)*64+PPOINT(X,Y+2)*32+PPOINT(X,Y+3)*16+PPOINT
(X,Y+4)*8+PPOINT(X,Y+5)*4+PPOINT(X,Y+6)*2+PPOINT(X,Y+7)
3150 IF X=255 THEN A=0:GOTO 3170
3160 PRINT £-2,CHR$(A);:NEXT
3170 PRINT £-2,CHR$(10);
3180 Y=Y+8:IF Y<191 THEN 3110
3190 PRINT £-2,CHR$(27);"2";CHR$(18);
3200 PRINT £-2:PRINT £-2,TAB(5)"PLOTTED USING EQUIVALENT CARTESIAN CO-ORDINATES
(R*COS(A*Z),R*SIN(B*Z))"
3210 PRINT:PRINTTAB(7)"PRINTING COMPLETED"
3220 PRINT:PRINTTAB(2)"PRESS REQUIRED KEY  (fslpdhte)"
3230 RETURN
3240 REM VIEW DIRECTORY
3250 CLS
3260 PRINTTAB(9)"DISK DIRECTORY"
3270 PRINTTAB(9)STRING$(14,45):PRINT
3280 GOSUB 2540
3290 DIR:IF PEEK(1462)<>ASC("M")  AND P<>5 THEN 3310 ELSE IF PEEK(1462)<>ASC("M")
 AND P=5 THEN 3340
3300 K$=INKEY$:IF K$=""THEN 3300 ELSE 3290
3310 PRINT@482,"PRESS THE SPACEBAR TO PROCEDE";
3320 K$=INKEY$:IF K$=""OR K$<>CHR$(32)  THEN 3320
3330 CLS:RETURN
3340 PRINT@482,"PRESS REQUIRED KEY  (FSLPDHTE)";
3350 RETURN
3360 REM TUTORIAL
3370 PMODE 4,1:SCREEN 1,1:PCLS1:COLOR 0,1
3380 SC=4:GOSUB 1090:AN=0:CO=0
3390 Y=1:XS=5:YS=10:SC=4:AN=0:CO=0
3400 M$="POLAR COORDINATES - TUTORIAL"
3410 GOSUB 1110:GOSUB 1150:GOSUB 1130:X=0:Y=11
3420 M$="Polar Coordinates is a method by which the locationof a given point P i
n the plane may be defined.  Wecommence with one Axis and a point on it called t
hePole."
3430 GOSUB 1150
3440 LINE(85,176)-(170,176),PSET
3450 FOR T=1 TO 5
3460 CIRCLE(85,176),6,0,1,0.04,0.96:WAIT 150
3470 CIRCLE(85,176),6,1,1,0.04,0.96:WAIT 150
3480 NEXT:X=X+(XS*2)
3490 M$="A point in the plane is now represented by apair of numbers (R,Z), wher
e R denotes its distancefrom the Pole"
3500 GOSUB 1150:TX=X:TY=Y
3510 X=152:Y=127:M$="(R,Z)":GOSUB 1150
3520 X=114:Y=142:M$="R":GOSUB 1150:X=TX+XS:Y=TY
3530 FOR T=1 TO 5
3540 LINE(85,176)-(150,130),PRESET:WAIT 150
3550 LINE(85,176)-(150,130),PSET:WAIT 150
3560 NEXT
3570 M$="and Z is the angle formed between theAxis and the line from the Pole to
 this point, thisangle being measured anticlockwise if Z is positiveor clockwise
 if Z has a negative value."
3580 GOSUB 1150:TX=X:TY=Y
3590 X=103:Y=166:M$="Z":GOSUB 1150:X=TX+(XS*2):Y=TY
3600 FOR T=1 TO 5
3610 CIRCLE(85,176),10,1,1,0.90,1:WAIT 150
3620 CIRCLE(85,176),10,0,1,0.90,1:WAIT 150
3630 NEXT
3640 M$="However, Ris always taken as positive.  We write P^(R,Z), thesymbol '^'
 standing for 'is uniquely defined by'."
3650 GOSUB 1150:GOSUB 3660:GOTO 3680
3660 Y=181:M$="PRESS THE SPACEBAR TO CONTINUE":GOSUB 1110:GOSUB 1150:GOSUB 1130
3670 K$=INKEY$:IF K$=""OR K$<>CHR$(32) THEN 3670 ELSE RETURN
3680 GOSUB 3690:GOTO 3710
3690 PCLS1:Y=1:M$="TUTORIAL (CONTINUED)":GOSUB 1110:GOSUB 1150:GOSUB 1130
3700 X=0:Y=11:RETURN
3710 M$="It will be seen that if either or both of these twocoordinates are chan
ged, then the location of pointP will also change accordingly, as in the followi
ngexamples:-"
```

```
3720 GOSUB 1150:Y=Y+YS:X=7
3730 LINE(120,120)-(190,120),PSET:WAIT 150
3740 M$="P1^(2.5,35)":GOSUB 3860
3750 LINE(120,120)-(172,84),PSET
3760 X=174:Y=80:M$="P1":GOSUB 3870
3770 M$="P2^(1.8,130)":GOSUB 3860
3780 LINE(120,120)-(92,86),PSET
3790 X=86:Y=80:M$="P2":GOSUB 3870
3800 M$="P3^(3,220)":GOSUB 3860
3810 LINE(120,120)-(62,169),PSET
3820 X=56:Y=166:M$="P3":GOSUB 3870
3830 M$="P4^(2.3,-40)":GOSUB 3860
3840 LINE(120,120)-(164,157),PSET
3850 X=166:Y=153:M$="P4":GOSUB 3870:GOTO 3880
3860 GOSUB 1150:GOSUB 1130:TX=X:TY=Y:RETURN
3870 GOSUB 1150:X=TX+XS:Y=TY:WAIT 1000:RETURN
3880 GOSUB 3660:GOSUB 3690
3890 M$="A function that involves Polar Coordinates (R,Z) iscalled a"
3900 GOSUB 1150:X=X+XS
3910 M$="POLAR FUNCTION.":GOSUB 1150:GOSUB 1130:X=X+(XS*2)
3920 M$="For instance, R=SIN(Z) isa Polar Function.  Substituting this example i
n ourbasic formula, P^(R,Z), would give us P^(SIN(Z),Z)."
3930 GOSUB 1150:X=0:Y=Y+YS
3940 M$="To draw the graph of a Polar Function, we take eachvalue of Z in some s
pecified range, and use this toplot the point P^(R,Z) using Polar Coordinates."
3950 GOSUB 1150:X=0:Y=Y+(YS*2)
3960 M$="To make the plotting easier the program substitutes"
3970 GOSUB 1150:X=0
3980 M$="CARTESIAN COORDINATES":GOSUB 1150:GOSUB 1130:X=X+XS
3990 M$="in its calculations, which isthe more usual method of representing poin
ts in theplane.  The point (R,Z) in Polar Coordinates is theequivalent of (R*COS
(Z),R*SIN(Z)) in Cartesian Coo-rdinates, and this is what we plot."
4000 GOSUB 1150:GOSUB 3660:GOSUB 3690
4010 M$="Many interesting & complex patterns can be producedif instead of using
the basic Cartesian Coordinates(R*COS(Z),R*SIN(Z)) we now introduce two addition
alnumbers, (A,B), & plot (R*COS(A*Z),R*SIN(B*Z))."
4020 GOSUB 1150:X=X+(XS*2)
4030 M$="Ifboth A & B are given a value of 1, the program willcalculate & displa
y the STANDARD PLOT of the chosenfunction.  Assigning any other values to A and/
or Bwill result in some quite spectacular and beautifulvariations."
4040 GOSUB 1150
4050 Y=150:SC=8:GOSUB 1090
4060 M$="TUTORIAL ENDED":GOSUB 1110:GOSUB 1150:GOSUB 1130
4070 Y=180:SC=4:GOSUB 1090
4080 M$="PRESS REQUIRED KEY (FSLPDHTE)":GOSUB 1110:GOSUB 1150:GOSUB 1130
```

```
4090 RETURN
4100 REM FORMULAE
4110 DATA SIN(2*Z)
4120 DEF FNA(Z)=SIN(2*Z):RETURN
4130 DATA SIN(5*Z)
4140 DEF FNA(Z)=SIN(5*Z):RETURN
4150 DATA SIN(6*Z)
4160 DEF FNA(Z)=SIN(6*Z):RETURN
4170 DATA SIN(7*Z)
4180 DEF FNA(Z)=SIN(7*Z):RETURN
4190 DATA SIN(9*Z)
4200 DEF FNA(Z)=SIN(9*Z):RETURN
4210 DATA 1+SIN(2*Z)
4220 DEF FNA(Z)=1+SIN(2*Z):RETURN
4230 DATA 1+SIN(5*Z)
4240 DEF FNA(Z)=1+SIN(5*Z):RETURN
4250 DATA 1+SIN(6*Z)
4260 DEF FNA(Z)=1+SIN(6*Z):RETURN
4270 DATA 1+SIN(7*Z)
4280 DEF FNA(Z)=1+SIN(7*Z):RETURN
4290 DATA 1+SIN(9*Z)
4300 DEF FNA(Z)=1+SIN(9*Z):RETURN
4310 DATA 1+COS(Z)
4320 DEF FNA(Z)=1+COS(Z):RETURN
4330 DATA 1+2*COS(Z)
4340 DEF FNA(Z)=1+2*COS(Z):RETURN
4350 DATA Z/4
4360 DEF FNA(Z)=Z/4:RETURN
4370 DATA 1+2*COS(2*Z)
4380 DEF FNA(Z)=1+2*COS(2*Z):RETURN
4390 DATA 1+2*COS(5*Z)
4400 DEF FNA(Z)=1+2*COS(5*Z):RETURN
4410 DATA 1+2*COS(7*Z)
4420 DEF FNA(Z)=1+2*COS(7*Z):RETURN
```

Continued from page 21
providing extended forwards and backwards paging.

Lines 2170-2240 I needed a way of labelling the graphs for disc storage and retrieval. Using the actual formulae was not possible as they contain more than eight characters. Likewise, trying to invent descriptive filenames proved to be a non-starter. Using identity numbers was the only viable alternative I could think of.

Line 2460 If the standard, as opposed to an enhanced, plot of a function has been drawn, this line PSETS a single, unobtrusive pixel, so that when the graph is printed, Line 3040 will print the appropriate heading.

Lines 2910-3230 Hard copy if required. printer instructions are for a Brother HR-5 printer. Alter the screen dump commencing at line 3090 to suit your own printer. If you have a machine code routine, so much the better. Better still, if you have a copy of MacGowan's *Dumper* program, you could save a copy of this on the same disc and re-write this routine to call it up from within the main program when a screen dump was required.

Lines 3250-3350 Access the disk directory. A single-drive system has been assumed, but the routine can easily be altered to suit a twin-drive system if required. My own system has a SuperDos E-6 controller, which allows the directory to be viewed a screenful at a time, indicating that extra items are to follow by printing the word MORE towards the bottom-right corner of the screen. By PEEKing at this location (line 3290) to ascertain whether the first letter of MORE is present or not, the program either loops back to the next page of the directory or, according to the value of P, returns to the current subroutine or to the Help screen. If your own system has a different controller, this routine will probably be of no use to you in its present form, and will require re-writing as necessary.

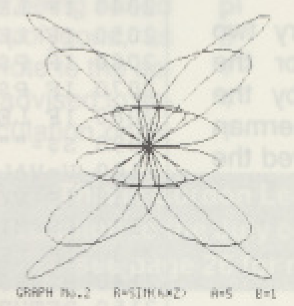
Lines 3370-4090 Tutorial. It is essential that the print string$ are typed in exactly as shown, otherwise the text will not be printed correctly.

Lines 4110-4420 Formulae. These lines are paired. The first lines of each pair are the data statements which are read by line 1290 for display in the functions library. The second lines contain the equivalent defined functions for use in the actual plotting. Incidentally, does anyone know of a method by which one can line input defined functions directly while a program is running?

If the thought of that typing dismays you, I can supply copies of the program at the following rates, inclusive of postage and packaging; on disc supplied by me £5.00; on disc supplied by purchaser £3.00; on cassette supplied by me £4.00; on cassette supplied by purchaser £3.00. Cheques, etc. to; K. Redhead, 21 Baxter Avenue, Newcastle upon Tyne NE4 9QD.

Finally, the program has been carefully tested and is bug-free to the best of my belief. However, if anyone with greater expertise than myself would like to write in with suggestions for improving my programming techniques, I am willing to learn and thank you in advance.

PLOT OF POLAR FUNCTION (ENHANCED)



PLOTTED USING EQUIVALENT CARTESIAN CO-ORDINATES (P*COS(A*Z),R*SIN(B*Z))

CLASSIFIED ADS

TWO double sided double density 80 track drives (1 unit) with double density DeltaDOS £180. Centronics GLP A4 printer £75. GEC wide carriage dot matrix printer £75. Many books, DU magazines, discs, tapes, cartridges etc. Telephone (0582) Luton 867527 evenings.

DRAGON 32, Dragon disc drive, Dragon 64, double Dragon disc drive, SuperDOS with accessories in original boxes, software, games, utilities, compilers, spreadsheets, books, mags, manuals. £400. Will split. Call 049161 3247 Oxfordshire.

FINAL CHANCE SALE, Dragon 32 £20, no pets allowed in new house. SAE Full list, Nic Spiers, 114 Greenway, Tunbridge Wells, Kent TN2 3JN.

HERE'S MY CLASSIFIED AD.
(please write your copy in capitals on the lines below)

Name . . . .
Address . . . .
. . . . **Tel:**

Classified rate: 35p per word.
Please cut out and send this form to: Classified Department, Dragon User, 49 Alexandra Road, Hounslow, Middx TW3 4HP

Communications

Write down your problem on the coupon below (make it as brief and legible as possible) together with your name and address and send it to Communication, 49 Alexandra Road, Hounslow, Middx TW3 4HP.

Problem . . . .

. . . .

. . . .

. . . .

Name . . . .

Address . . . .

. . . .

. . . .

Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpine, 49 Alexandra Road, Hounslow, Middx TW3 4HP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure . . . .

Problem . . . .

. . . .

Name . . . .

Address . . . .

. . . .

A nice round sum

So mysterious is pi that it took 1600 years to find 35 digits

LAST month, we commenced a brief examination of the mysteries of the number pi, the ratio of a circle's circumference to its diameter. To conclude our review of this number let's look at some of the attempts that have been made to calculate its value to greater degrees of accuracy. As an indication of its enigmatic nature it was not until the sixteenth century that its value was known with certainty to ten decimal places. This was the result of an amateur mathematician, Francisco Vieta, who also devised one of the first methods of calculating pi by means of a non-terminating series. The close of the sixteenth century saw the accuracy extended to 35 decimal places by the German/Dutch mathematician Ludolph van Ceulen. At his own request, these 35 digits were engraved on his tombstone.

During the seventeenth century two more methods were devised for the calculation of pi. These were by the Englishman John Wallis and the German G W Liebniz. Both methods involved the use of non-terminating series:

$$\frac{pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \frac{8}{9} \quad \ldots$$

$$\frac{pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \quad \ldots$$

The first of these (Wallis') consists of an endless stream of products, while the second is a sum with alternating positive and negative components. By continuing the series indefinitely the value of pi can, in theory, be calculated to as many decimal

```
100 CLEAR 3000:OP=1:L$="5":M$="239":R=3
110 A$=L$:GOSUB 2000:J$=Q$:A$=M$:GOSUB 2000:K$=Q$
120 Z$=L$:M=25:GOSUB 1000:L$=P$:Z$=M$:M=57121:GOSUB 1000:M$=P$
130 Z$=L$:M=R:GOSUB 1000:A$=P$:GOSUB 2000:N$=Q$
140 Z$=M$:M=R:GOSUB 1000:A$=P$:GOSUB 2000:O$=Q$
150 A$=MID$(J$,3):B$=MID$(N$,3):ON OP GOSUB 4000,3000:J$="0."+A$
160 A$=MID$(K$,3):B$=MID$(O$,3):ON OP GOSUB 4000,3000:K$="0."+A$
170 Z$=K$:M=4:GOSUB 1000:B$=P$:Z$=J$:M=16:GOSUB 1000:A$=P$
180 GOSUB 4000:PRINT"R.F.= ";R:PRINT"3.";MID$(A$,3)
190 IF OP=1 THEN OP=2 ELSE OP=1
200 R=R+2:GOTO 120

1000 P$="":C=0:FOR L=LEN(Z$)TO 1 STEP-1
1010 V=VAL(MID$(Z$,L,1)):V=V*M+C:V$=MID$(STR$(V),2):C=0
1020 P$=RIGHT$(V$,1)+P$:C=VAL(LEFT$(V$,LEN(V$)-1))
1030 NEXT:IF C<>0 THEN P$=MID$(STR$(C),2)+P$
1040 RETURN

2000 Q$="0."+STRING$(LEN(A$)-1,"0")
2010 B$="1"+STRING$(LEN(A$),"0")
2020 M=1
2030 Z$=A$:GOSUB 1000
2040 IF LEN(B$)<LEN(P$)THEN B$="0"+B$:GOTO 2040
2050 IF LEN(P$)<LEN(B$)THEN P$="0"+P$:GOTO 2050
2060 IF P$<B$ THEN M=M+1:GOTO 2030
2070 IF P$=B$ THEN 2090 ELSE M=M-1:GOSUB 1000
2080 IF LEN(P$)<LEN(B$)THEN P$="0"+P$:GOTO 2080
2090 S$="":C=0:FOR G=LEN(P$)TO 1 STEP-1
2100 V=VAL(MID$(B$,G,1))-VAL(MID$(P$,G,1))-C:C=0
2110 IF V<0 THEN V=V+10:C=1
2120 S$=MID$(STR$(V),2)+S$:NEXT
2130 IF LEFT$(S$,1)="0"THEN S$=MID$(S$,2):GOTO 2130
2140 B$=S$+"0":Q$=Q$+MID$(STR$(M),2):IF LEN(Q$)<252 THEN GOTO 2020
2150 RETURN

3000 T$="":C=0:FOR G=LEN(A$) TO 1 STEP-1
3010 V=VAL(MID$(A$,G,1))+VAL(MID$(B$,G,1))+C:C=0
3020 IF V>9 THEN V=V-10:C=1
3030 T$=MID$(STR$(V),2)+T$:NEXT:A$=T$:RETURN

4000 T$="":C=0:FOR G=LEN(A$) TO 1 STEP-1
4010 V=VAL(MID$(A$,G,1))-VAL(MID$(B$,G,1))-C:C=0
4020 IF V<0 THEN V=V+10:C=1
4030 T$=MID$(STR$(V),2)+T$:NEXT:A$=T$:RETURN
```

Prize

ASSUMING that you take less than 1600 years to work out The solution to Gordon Lee's puzzle, and don't require a custom-made headtone at the end of your labours, you will be in line for one of ten copies of *Indoor Football*, by kind permission of **Computape**. This ex-Quickbeam game has been given league-beating reviews in DU.

Rules

KICK the idea about a bit on your Dragon until you get a-round to an answer. Park it on paper (no tapes this time, please), with any additional remarks (remember, this is sone of the few competitions which allows correspondence, at least before the judging), put down your name and address, and send it off to us in an envelope marked OCTOBER COMPETITION.

For the tiebreaker, use your wit and wisdom to complete the phrase "I like Indoor Football better than outdoor football because ...".

July winners

QUESTION: what have Dragon users in common with God, apart from a tenacious commitment to project they said would never last? Answer: they move in a mysterious way. We had lots of entries from first-timers this time. Why? How do I know? But welcome aboad,

This month's winners are David Marsden of Garston (seriously), Ton Denton of South Norwood, Denis O'Mulloy of Cambridge, John Blatch of Addlestone, P D Maddocks of Taplow, Malcolm McBride of Little Harrowden, Roy Cashmore of Blaston, Fred Willers of Yarnfield and Ron Raine of Sapcote.

These ten will be receiving copies of Super Nova from Orange Software. They will, that is, when I find the list of names and addresses that Gordon so kindly sent me. I have just absorbed a brilliant article called How To Manage Your Paper, which I am told will change my life. The gist of it is that you should throw everything away. Hold on, perhaps I . . .

Anyway, favourite tiebreaker from D J Gray: "My wife says, my favourite number is umpteen, because that's the number of times she has to tell me to do anything. But actually my favourite number is umpteen minus one, because that is the only command I actually hear." (Try "Dearest, there's a man at the door who wants to give you a free Maserati". He'll hear that.) And countless members of our community had favourite ages that they will never see again. Don't worry — according to the law of recurring series, there should be another one along soon.

Solution

See opposite page

places as is desired. These two formulae have been converted into the listings given here in which variables S, D, and N (where appropriate) represent the step number, denominator and numerator respectively. Before the euphoria sets in at finding a method of calculating pi, I should point out that in a practical sense, both of these are strictly limited, as will be seen if these listings are typed in and run.

A far more useful formula is that known as the Machin Formula as it resolves much more speedily than the two already given:

$$= 16\left(\frac{1}{5} - \frac{1}{3} \times \frac{1}{5^3} + \frac{1}{5} \times \frac{1}{5^5} - \ldots\right)$$

$$- 4\left(\frac{1}{239} - \frac{1}{3} \times \frac{1}{239^3} + \frac{1}{5} \times \frac{1}{239^3} - \ldots\right)$$

This formula can be adapted into a listing in a similar manner to the others and will quickly evaluate pi to several decimal places, however, because of limitations in the arithmetic capacity of the Dragon, an overflow (OV) error will soon be encountered as a result of raising 239 to a power greater than 15.

While writing this account of pi I came across a short account in a century old book on the calculation of pi correct to 6 or

```
WALLIS

10 X=2:S=1:N=2:D=3
20 Z=2*X
30 PRINT S;"   ";Z
40 X=X*N/D
50 IF S/2=INT(S/2)THEN D=D+2
   ELSE N=N+2
60 S=S+1
70 GOTO 20

LEIBNIZ

10 X=1:S=1:D=3
20 Z=4*Z
30 PRINT S;"   ";Z
40 F=1/D
50 IF S/2=INT(S/2)THEN X=X+F
   ELSE X=X-F
60 S=S+1:D=D+2
70 GOTO 20
```

7 figures. The book was *A Treatise on Elementary Trigonometry* by the Rev. J B Rock, and the method used was a very intricate method of arithmetically calculating square roots to an accuracy of about 8 decimal places (the method being based on Archimedes' inscribed polygons, as outlined last month). A footnote to the account says: "The student is advised to actually work through the calculation once. and he will have the satisfaction of having himself calculated the value of pi." By adapting the Machin Formula and utilising the 'string' arithmetic method of calculation referred to a number of times on this page recently, the second listing will carry out this evaluation, and will, as J B Lock remarked a century ago, give the satisfaction (albeit using a computer) of having calculated the value of pi.

For the competition this month we are returning to the two formulae by Wallis and Leibniz. The disadvantage, as will be readily seen, is that the value of pi is computed exceedingly slowly. For example, using Wallis' formula it takes nine steps before the first digit of pi — the 3 — stabilises. The next digit, the 1, does not stabilise until the 37th step, while it takes 985 steps before the third digit — the 4 — is known with certainty. The table below lists the first three results for both of these formulae. Can you fill in the correct figure for the fourth digit?

pi		Wallis	Leibniz
3	step	9	7
1	step	37	25
4	step	985	627
1	step	?	?

The Answer

This is Gordon Lee's own solution to the July competition see page 26 for results

THE listing given here will give a reasonable result for numbers in the range 1 to 999999999. Because of the way in which certain numbers are phrased when spoken, certain checks are carried out in the program — in particular, to determine if an 'and' needs to be inserted in order that the result sounds correct. For example, we say 'one thousand *and* one', but we drop the 'and' if we say 'one thousand four hundred'.

When dealing with a nine digit number we can fortunately employ a number of short cuts. For instance the nine-digit number can be split into three three-digit segments, such that we speak of 'so many million, so many thousand and so many'. Each of the three digit segments can be handled in the same way. This is done in the subroutine from line 2000. Each of the three digits is extracted into variables D1, D2, and D3. D1 will be the number of 'hundreds' in the triplet. If this is zero the step is skipped, otherwise the relevant word from the array D$ is inserted into string W$. Similarly, the digits representing the tens is done in the same way except that the array T$ is used which holds the terms 'Twenty', 'Thirty', 'Forty' and so on. The units are done in a similar manner. The only irregularity is in handling the 'teens'. Thus, all values less than 20 are handled as though they were units (line 2210) and are read from the array D$ — i.e. 13 would be read as 'thirteen'.

The remainder of the program relates to putting together the three segments representing the millions, thousands, and units (units here meaning values under one thousand). Once again, checks are carried out to determine where an 'and' should be placed rather than just a space.

```
>
  10 DIM D$(19),T$(9)
  20 FOR F=1 TO 19:READ Z$
  30 D$(F)=Z$:NEXT
  40 FOR F=1 TO 9:READ Z$
  50 T$(F)=Z$:NEXT
 100 INPUT Q$
 110 IF LEN(Q$)<9 THEN Q$="0"+Q$:GOTO 110
 120 M$=LEFT$(Q$,3):T$=MID$(Q$,4,3):U$=RIGHT$(Q$,3)
 130 Z$=""
 140 IF M$="000"THEN 200
 150 N$=M$:GOSUB 2000
 160 Z$=Z$+W$+" MILLION "
 170 IF T$="000" AND U$="000"THEN 340
 200 IF T$="000"THEN 300
 210 N$=T$:GOSUB 2000
 220 Z$=Z$+W$+" THOUSAND"
 230 IF U$="000"THEN 340
 300 IF M$="000"AND T$="000" THEN 320
 310 IF T$="000" OR LEFT$(U$,1)="0" THEN Z$=Z$+" AND " ELSE Z$=Z$+" "
 320 N$=U$:GOSUB 2000
 330 Z$=Z$+W$
 340 PRINT Z$:GOTO 100
 999 END
 1000 DATA ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,ELEVEN,TWELVE,THIRTE
EN,FOURTEEN,FIFTEEN,SIXTEEN,SEVENTEEN,EIGHTEEN,NINETEEN
 1010 DATA TEN,TWENTY,THIRTY,FORTY,FIFTY,SIXTY,SEVENTY,EIGHTY,NINETY
 2000 W$=""
 2010 IF LEN(N$)<3 THEN N$="0"+N$:GOTO 2010
 2020 D1=VAL(MID$(N$,1,1))
 2030 D2=VAL(MID$(N$,2,1))
 2040 D3=VAL(MID$(N$,3,1))
 2100 IF D1=0 THEN 2200
 2110 W$=W$+D$(D1)+" HUNDRED"
 2120 IF D2=0 AND D3=0 THEN 2500
 2130 W$=W$+" AND "
 2200 IF D2=0 THEN 2300
 2210 IF D2*10+D3<20 THEN W$=W$+D$(D2*10+D3):GOTO 2500
 2220 W$=W$+T$(D2)
 2230 IF D3=0 THEN 2500
 2240 W$=W$+" "
 2300 IF D3=0 THEN 2500
 2310 W$=W$+D$(D3)
 2500 RETURN
```

Dragon Answers

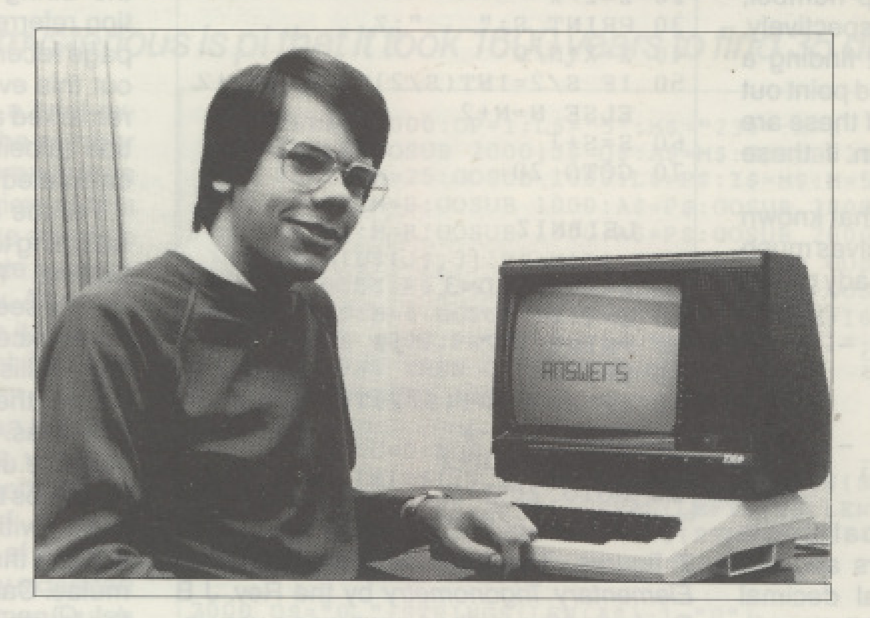
If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.

Seek for the Answer

I own a Tandy 64 with disc drive and OS-9. Reading the OS-9 manual, it mentions the possiblity of using tapes. This would be ideal for quick backups and may be the chance of communicating with another Dragon while running OS-9.

C. Moss
63 Withycombe Road
Penketh
Cheshire
WA5 2QJ

WHAT you need is an OS-9 device driver for the cassette system. One appeared in the March 1988 issue of DU. You'll need an assembler and a fair knowledge of OS-9 to use it. Alternatively, perhaps someone knows of abailable drivers for OS-9 (public domain?) and will let us know.



Pin No.	Dragon	IBM
10	Drive 1 Select	Motor Enable A
12	Drive 2 Select	Drive Select B
14	-	Drive Select A
16	Motor On	Motor Enable B

IBM clone

I have been thinking of getting my Dragon 32 a disc system. I have a spare IBM PC floppy drive of 360K and I have been wondering which disc controller I will need to use it with the Dragon. Can you tell me which Dragon specific operating system is the closest to IBM PC-DOS?

John Edwards
32 Chesham Road
Norton
Cleveland

THE pin connections of a PC drive are almost identical to those specified by DragonDOS (and therefore any compatible cartridge controller. That is, all odd numbered pins are Ground, pin 8 is 'index' through to pin 32 which is 'select head 1'.

The only differences are in pins 10, 12, 14 and 16:

You'll probably find that for a single drive no crossovers need be made. PNP manufacture suitable disc controller cartridges, which are available from Bob Harris at his usual address (check for price and availability). As far as Dragon OSs are concerned, it is a choice between FLEX and OS-9. Neither is much like MSDOS, but OS-9 is probably nearer in structure (stand by for the complaints from FLEX fanatics again!).

Every which way but Left . . .

COULD you please tell me if there is a fairly simple machine code routine for scrolling the text screen left, one column at a time?

I have routines for scrolling the screen in the other three directions, but I cannot complete the program I am writing without a left scroll.

U R Salthouse
7 RIver Street
Ware
Herts
SG12 7AF

THE following routine will do the trick. It is relocatable, so just POKE the code (hex bytes given in left column) wherever you want and EXEC at that address.

```
          ;  SCROLL  TEXT LEFT
          ;
8E 04 00             LDX   £1024
C6 1F     LOOP1:     LDB   £31
A6 01     LOOP2:     LDA   1, X
A7 80                STA   , X+
5A                   DECB
26 F9                BNE   LOOP2
86 60                LDA   £96
A7 80                STA   , X+
8C 06 00             CMPX  £1536
25 EE                BLO   LOOP1
39                   RTS
```

OS-9 driver

I own a Dragon 32 computer with a Dragon Data disc drive and recently a fault has developed in the DOS. When I use any command such as DIR, the drive indicator light comes on and a SKERROR code is reported. My manual doesn't explain the meaning or cause of this error.

Is this a fault with the controller or dirve and could it be rectified by buying a SuperDOS chip?

Sotos Mandalos
2 Kibblewhite Crescent
Twyford
Berks
RG10 9AX

THE 'SK' error stands for SeeK error. This occurs when the drive controller chip's request for a seek (or move) to one of the 40 tracks on a disc fails. This could be due to a fault in the controller cartridge, or more likely in the drive itself. Occasionally a dirty edge connector is to blame (try removing the cartridge and cleaning the contacts with isopropyl alchohol, available from the pharmacy counter at chemists).

If this has no effect, you'll need to get your drive serviced. In any case, replacing the DragonDOS eprom with a SuperDOS eprom will not help.

Recurring DREAMs

FINALLY, a note on the problems of saving source code from Dream. It seems many of you have the same problem and all say the various patches published do not work — well, it now seems that there are at least three totally different versions of Dream in circulation. The patches do work, but only on the most common version. Watch this space...